Predicting Multiple Target Tracking Performance for Applications on Video Sequences

Juan Esteban Tapiero Bernal
Marquette University

PREDICTING MULTIPLE TARGET TRACKING PERFORMANCE FOR APPLICATIONS ON VIDEO SEQUENCES

by

Juan E. Tapiero Bernal, B.S., M.Sc.

A Dissertation Submitted to the Faculty of the Graduate School,
Marquette University,
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy

Milwaukee, Wisconsin

August 2016

ABSTRACT

PREDICTING MULTIPLE TARGET TRACKING PERFORMANCE
FOR APPLICATIONS ON VIDEO SEQUENCES

Juan E. Tapiero Bernal, B.S., M.Sc.
Marquette University

This dissertation presents a framework to predict the performance of multiple target tracking (MTT) techniques. The framework is based on the mathematical descriptors of point processes, the probability generating functional (p.g.fl). It is shown that conceptually the p.g.fls of MTT techniques can be interpreted as a transform that can be marginalized to an expression that encodes all the information regarding the likelihood model as well as the underlying assumptions present in a given tracking technique. In order to use this approach for tracker performance prediction in video sequences, a framework that combines video quality assessment concepts and the marginalized transform is introduced. The multiple hypothesis tracker (MHT), Joint Probabilistic Data Association (JPDA), Markov Chain Monte Carlo (MCMC) data association, and the Probability Hypothesis Density filter (PHD) are used as a test cases. We introduce their transforms and perform a numerical comparison to predict their performance under identical conditions. We also introduce the concepts that present the base for estimation in general and for applications in computer vision.

ACKNOWLEDGEMENTS

Without the guidance, patience, and encouragement of countless people during my education, the completion of this dissertation would not have been possible. Thus, I would like to express my deepest thanks to those who have contributed to the process and helped me to reach this point.

First, I would like to extend my gratitude to Dr. Robert Bishop for providing me with the opportunity to further my studies and pursue a doctorate degree. His guidance and feedback throughout the past few years has been invaluable to my research.

I also owe my thanks to Dr. Henry Medeiros for exchanging ideas with me. His input and advice helped to direct and ground the ideas in my dissertation.

In addition to Dr. Medeiros, his graduate students, Andrs F. Echeverri Guevara and Tony Hoak, contributed their perspectives and helped me to problem-solve. They also afforded me the opportunity to explain my work and practice speaking about it professionally.

I must also thank my dissertation committee, which includes Dr. Elaine Spiller, Dr, Richard Povinelli, Dr. Robert Bishop, Dr. Edwin Yaz, and Dr. Henry Medeiros. Their assistance and attention to my drafts has been greatly beneficial to the revision process.

I truly appreciate the Vizlab at Marquette for helping to develop a digital world in order to test my algorithms.

Additionally, Dr. Yaz assisted me by helping with logistics such as university-related paperwork, which I am grateful for.

Finally, I am indebted to my family and friends for all of their unconditional support and belief in me during all of my schooling. I thank my girlfriend Marissa for proofreading my drafts and preparing some of my chapters. I would especially like to thank my parents for always encouraging my intellectual pursuits and working to grant me the best opportunities they could. This work is dedicated to my parents and my loving girlfriend.

TABLE OF CONTENTS

TABLE OF CONTENTS — *Continued*

TABLE OF CONTENTS — *Continued*

LIST OF TABLES

LIST OF FIGURES

LIST OF NOMENCLATURE AND ACRONYMS

MTT: Multiple Target Tracking.

MHT: Multiple Hypothesis Tracker.

JPDA: Joint Probabilistic Data Association.

MCMC: Markov Chain Monte Carlo.

RFS: Random Finite Sets.

PHD: Probability Hypothesis Density.

GMMPHD. Gaussian Mixture Model Probability Hypothesis Density.

PPP: Poisson Point Process.

BMD: Bayes-Markov Filters with Miss Detection.

TQA: Tracker Quality Assessment.

OSPA: Optimal Subpattern Assignment metric.

Table 0.1: Nomenclature

math	definition
boldface	vector or Matrix
italic	vector magnitude or scalar
lowercase	vector
UPPERCASE	Matrix or set
$\odot$	dot product
$p(\bullet)$	Probability density function
$\mathcal{N}(\bullet)$	Normal Distribution
$\Psi[\bullet]$	Probability generating functional
$\| \bullet \|$	standard Euclidean norm of a vector
$\| \bullet \|$	Determinant of a matrix
n	number of states, or number of targets. Context dependent
m	number of measurements

CHAPTER 1

Introduction

Solutions to the problem of multiple target tracking are comprised of many techniques and algorithms that generalize the single target tracking and state estimation for more complex tracking scenarios. Since a unified theory for single target tracking and estimation does not exist, this set of techniques has a growing number of elements that contribute to the solutions of the problem of multiple target tracking under different assumptions and approaches which likely involve Bayesian probabilities. Today, most used techniques can be classified in one of four frameworks:

1 Extension of the Bayesian framework for single target tracking [1]. Classical examples are multiple hypothesis tracking, joint probabilistic data association (JPDA) filter, and Markov Chain Monte Carlo framework.

2 Random finite sets framework [2]. Examples include Probability Hypothesis Density filters, Cardinalized Probability Hypothesis Density filters, and Multi-Bernoulli filters.

3 Point processes framework [3]. Examples include the intensity filter.

4 Heuristic implementations such as the nearest neighbor (NN) standard filter [4].

There are more examples in each of the four frameworks and the number keeps growing. Much of the new developments represent small variations with some conceptual progress on solving the problem of data association.

Computer vision and specifically pedestrian detection is one of the most widely investigated problems in which multiple target tracking is applied since it has an impact on practical matters such as robotics, surveillance, video game industry and sports broadcasting [5], and with the available computation of the day offers great experimenting flexibility and ways to evaluate different algorithms, and video sequences perse are self-contained and do not require extra information or structures.

There is not a structured approach to select the appropriate estimation technique to be applied on a video sequence, but different techniques can be tested and compared through existing metrics of characteristics such as tracking accuracy and robustness [6] [7] [8] [9] [10]. The choice of the technique to be used on any given application is usually made on an ad-hoc basis following from targeted research based on the knowledge the designer has of certain techniques, or benchmarking using the formerly mentioned metrics applied to the results of multiple target tracking problem at hand [11] [12] [13]. The main questions are, could there be something about the problem than can point us towards the selection of one of the techniques as being preferable over all the others? Can the performance of the selected technique be predicted?

1.1 Objectives and Scope

This work aims to answer the posed question in the specific case of predicting the performance of Multiple Target tracking techniques for video sequences. For this it is important to introduce several sets of concepts introduce tracking techniques and that show the fundamental relationship between them and how knowledge about the objective video quality mixed with this fundamental relationships can be used as a framework for

performance prediction.

Given the large number of possible techniques thus the impossibility (in terms of time frame) of showing the details of all of them, we focus on four of the multiple target tracking techniques dubbed as the classics. Two of them are considered the original multiple target tracking techniques of the bayesian framework: multiple hypothesis tracker and joint probabilistic data association. The third selected technique is the Markov Chain Monte Carlo data association filter which extends on the classic JPDA for an unknown number of targets. Finally, one of the first filters that was introduced with the birth of the random finite sets framework, the probability hypothesis density (PHD) filter.

It is important also to note that the labelling and re-identification of targets is not taken in account, since it usually does not depend on probabilistic techniques to create the labels or multiple sensors (or cameras in this case), but on heuristic or added optimization structures. Another way of limiting the scope comes from the assumption that throughout the implementations and experiments the motion and measurement models are the mean of a Gaussian distribution.

1.2 Dissertation Contributions

Given the objective and questions posed, the main contributions of this dissertation are:

1 A new conceptualization of probability generating functionals for multiple target tracking as a marginalized tracking transform that encodes all the information contained within the measurements.

2 The introduction of a framework that uses concepts from visual quality assessment and

the newly introduced marginalized tracking transform to obtain a quantity called the tracker quality assessment to be used for multiple target tracking performance prediction.

3 Extensive testing of the introduced transformation and tracker quality assessment framework using different assumptions for the tracking techniques and different video sequences. Introducing a numerical analysis of what these new quantities mean for predicting multiple target tracking performance, both the marginalized transform and the tracker quality assessment framework.

1.3 Dissertation Outline

The second chapter has two main objectives, first it presents the general principles of bayesian estimation and then it presents all the pertinent concepts about modeling for computer vision applications. It is an extension of the introduction to present the general concepts of target tracking and estimation. Finally, the introduction of measurement models and techniques for detection of pedestrian on videos.

Chapter 3 presents the theoretical aspects of the general multiple target tracking as an extension of the single target problem, aiming to introduce a set of assumptions that change the way in which is problem is approached. Here we try for the most part to maintain a Bayesian way to present the different concepts and the techniques pertinent for this work, developing a framework that includes multiple hypothesis tracking, joint probabilistic data association filter and Markov Chain Monte Carlo data association. The basics of the probability hypothesis density filter are also discussed superficially.

Chapter 4 firstly introduces the general principles of point processes and then extends

this to probability generating functionals, introducing then their relationship with target tracking and presenting the representation of the different single target and most importantly multiple target tracking techniques in terms of probability generating functionals. A key contribution here comes from introducing the probability generating functional for the Markov Chain Monte Carlo data association framework. Finally, it presents the concepts that are used in this work to develop a marginalized tracker transform by re-interpreting the probability generating functionals.

Chapter 5 has a review of the subject of visual quality assessment for images and videos. In this chapter we introduce the framework that connects the Multiple Target Tracking transform concepts with visual quality assessment to obtain a quantity called: tracker quality assessment. This quantity is the key to predict tracker performance on videos.

Chapter 6 wraps up the work by describing the experiments performed and results obtained. It introduces too the implementations for the marginalized transform and analysis of their meaning, finally application and ways in which performance on videos can be predicted. Finally in Chapter 7 with this in hand, we make concluding remarks and point all the ways to improve and perform future research.

CHAPTER 2

Bayesian Estimation

This chapter introduces the general concepts that form the estimation framework and that are extended for the main subject of this dissertation and can be built upon to obtain a mathematically exact solution for the multiple target tracking problem. It also has a second purpose in the introduction of a proposed motion model for pedestrian tracking in computer vision applications, one of the contributions of this work.

2.1 Bayesian Inference

Bayesian inference is a theoretical, yet practical framework for reasoning, decision making and estimation under uncertainty. The historical roots of the theory lie in the late 18th and early 19th century with Thomas Bayes and Pierre-Simon de Laplace [14]. Bayesian inference was not a popular approach for decision making until the last half of the 20th century and did not develop as a single, homogeneous scientific activity. It has, however, been employed in many different domains. The Bayesian approach to filtering is not new (e.g., see Ho and Lee [14]; Jazwinski [15] ; Stratonovich [16]). The Kalman filter can be derived from the mean least-squares point of view or from a Bayesian perspective (see Kalman [17]). As computations became faster and more accessible, state estimators with a higher computational cost were developed. From these estimators we can consider three categories. First, the class of different extended Kalman filter (EKF) variants that provide estimates of the state variables and a measure of the mean least-square state estimation

error. In this category we can include the estimators that approximate the probability density function of the variable with a mixture of probability density functions. This was proposed by Sorenson and Alspach [18] by using a mixture of Gaussians. We might also consider grid based filters that evaluate the pdf using a series of nodes chosen to cover the entire state space. This set of nodes, each with an associated weight, are used as a discrete approximation of the posterior pdf or as base for continuous approximations for this pdf, for example using splines [19]. The last category of filters includes those that use *monte carlo* methods. Their origins can be traced to Handschin [20] and Mayne and Handschin [21]. Gordon et al [22] employ sequential *monte carlo* methods as set of points that approximate the posterior pdf. The objective of Bayesian estimation (in our application) is to estimate the state vector in a recursive form based on the discrete-time observations.

A scientific hypothesis is typically represented as a pdf of the observed data. This pdf depends on certain unknown quantities or parameters, denoted by θ. In the Bayesian paradigm, the knowledge of the model parameters is expressed through a pdf, known as the prior density function, $p(\theta)$. When new data y is obtained, the information contained in the prior pdf and its relation with the model parameters is known as the "likelihood" function, and is represented by $p(y|\theta)$. The information contained in the prior pdf and the likelihood function can be combined to obtain a new pdf, known as the posterior pdf and denoted by $p(\theta|y)$. The posterior pdf is the objective of the Bayesian inference process. Bayes theorem is an elemental identity in probability theory (more information on this and basic probability theory can be found in [23]). According to Bayes, the posterior probability is proportional to the product of the priori by the likelihood,

$$p(\theta|y) = \frac{p(\theta)p(y|\theta)}{\int p(\theta)p(y|\theta)\mathrm{d}\theta}.$$

In theory, one can always obtain the posterior distribution, but with the complex systems

and models the necessary analytical calculation are typically intractable. In recent years, the research community realized that obtaining samples of the posterior could be an applicable and adequate option.

There are several reasons to use Bayesian methods and their applications are present in several different fields. Many investigations into the use of Bayesian methods have been reported [24] [25] [26] [22] [27] [28] [29]. It is evident that if one wants to make a consistent decision in the presence of uncertainty, an excellent approach is to use Bayesian methods.

2.2 Single Target Bayesian Estimation

2.2.1 Continuous-Discrete Probabilistic Dynamical Systems

Probabilistic dynamical systems are a sequence of continuous probability density functions $p(\mathbf{x}(t_k)|\mathbf{y}_{1:k})$, where $\mathbf{x}(t_k)$ is the state vector, $\mathbf{y}_k$ is the observations vector, the index $t = t_k$ represents an instant of time when a observation is obtained and the subscript $1:k$ represents the set of observation at all instants up to and including t_k. The state variable $\mathbf{x}(t)$ evolves over time. In most of the applications, the difference between $p(\bullet|\mathbf{y}_{1:k})$ and $p(\bullet|\mathbf{y}_{1:k-1})$ is due to the incorporation of a new observation. The following processes are of special interest:

Prediction: $p(\mathbf{x}(t+dt)|\mathbf{y}_{1:k}), dt \neq 0$, where $p(\bullet|\mathbf{y}_{1:k})$ can be computed for all time $t > t_k$. The best prediction of $\mathbf{x}(t)$ before new information arrives is through $p(\bullet|\mathbf{y}_{1:k-1})$.

Smoothing: $p(\mathbf{x}(t)|\mathbf{y}_{1:T}), 0 < t < t_T$. In this case, the distribution can be calculated for all times $t \in [0, t_T]$ if the observations up to the instance $\mathbf{y}_T$ have been observed.

Estimation: $p(\mathbf{x}(t_k)|\mathbf{y}_{1:k})$ (Sequential estimation). Here we estimate the variable $\mathbf{x}(t_k)$ at the time instance t_k when the observation $\mathbf{y}_k$ has been obtained.

For this work, we consider dynamical systems that are represented in a state space form. A state space model is defined by the state equation,

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) + \sigma(\mathbf{x}(t), t)\mathbf{w}(t), \tag{2.2.1}$$

and the measurement equation,

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}(t_k), t_k, \mathbf{v}(t_k)) \tag{2.2.2}$$

where $\mathbf{y}_k$ is the observations vector at t_k, $\mathbf{x}(t)$ is the state vector, $\mathbf{h}$ is the measurement function (vector of functions), $\mathbf{f}$ is the state function (also known as the drift function), $\mathbf{u}(t)$ is the vector of inputs or control actions, $\mathbf{w}(t)$ is a stochastic noise process, $\mathbf{v}(t_k)$ is a random noise sequence, t is time, and t_k represents the instant an observation is obtained. The usual assumptions are that the analytical representation of the functions and distributions of both noises are known. The objective of Bayesian estimation is to estimate $\mathbf{x}(t_k)$ in a recursive form based on the observations $\mathbf{y}_k$, obtaining the posterior distribution $p(\mathbf{x}(t_k)|\mathbf{y}_{1:k})$.

State variables $\mathbf{x}$ and measurements $\mathbf{y}$ are directly related to the different probability density functions that represent the system when it is treated as a set of stochastic processes, and that are ultimately used for Bayesian estimation. In general, it can be said that

$$\mathbf{x} \sim p(\mathbf{x}(t_k)|\mathbf{x}(t_{k-1}))$$

$$\mathbf{y} \sim p(\mathbf{y}_k|\mathbf{x}(t_k)),$$

where $p(\mathbf{x}(t_k)|\mathbf{x}(t_{k-1}))$ is known as the transition density. There are two final definitions and

assumptions that are key to Bayesian estimation and inference. First, the Markov assumption which states that the values in any state $\mathbf{x}(t)$ are only influenced by the values of the state $\mathbf{x}(t-dt)$ that directly preceded it. This implies that the past is independent of the future. In a continuous-discrete setting, we have

$$p(\mathbf{x}(t_{0:k})) = \prod_{i=1}^{k} p(\mathbf{x}(t_i)|\mathbf{x}(t_{i-1}))p(\mathbf{x}(t_0)). \tag{2.2.3}$$

We also have the conditional independence of observations that states that the observation, $\mathbf{y}_k$, given the state, $\mathbf{x}(t_k)$, is conditionally independent from the observation and state history, or

$$\begin{aligned} p(\mathbf{y}_{1:k}) &= \prod_{i=1}^{k} p(\mathbf{y}_i) \\ p(\mathbf{y}_{1:k}|\mathbf{x}(t_{0:k})) &= \prod_{i=1}^{k} p(\mathbf{y}_i|\mathbf{x}(t_i)). \end{aligned} \tag{2.2.4}$$

2.2.2 Recursive Estimation

The well-known set of Bayesian filters are based on a general structure. Each filter differs under different assumptions. The main objective in each case is to estimate the state of a system using observations, where the state evolves in the presence of noise and observations are made sequentially in the presence of noise. The notation is as follows: $\mathbf{x}(t)$ is the state being estimated, and $\mathbf{y}_k$ indicates the observed data. The problem consists of estimating the state $\mathbf{x}(t_{0:k}), k = 1, 2, ...$ based on the sequence of observations $\mathbf{y}_{1:k}, k = 2, 3, ...$ In this derivation, the Markov assumption and conditional independence of observations assumption apply.

The set of posterior distributions can be represented using the Bayes theorem as

$$p(\mathbf{x}(t_{0:k})|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_{1:k}|\mathbf{x}(t_{0:k}))p(\mathbf{x}(t_{0:k}))}{p(\mathbf{y}_{1:k})}. \tag{2.2.5}$$

In a practical setting all the information needed to compute $p(\mathbf{x}(t_k)|\mathbf{y}_{1:k})$ is not known or cannot be obtained in real-time, so using the assumptions from Eqns. (2.2.3) and (2.2.4), we begin by rewriting Eq. (2.2.5) as

$$p(\mathbf{x}(t_{0:k})|\mathbf{y}_{1:k}) = \prod_{i=1}^{k} \frac{p(\mathbf{y}_i|\mathbf{x}(t_i))p(\mathbf{x}(t_i)|\mathbf{x}(t_{i-1}))p(\mathbf{x}(t_0))}{p(\mathbf{y}_i)}. \tag{2.2.6}$$

Eq. (2.2.6) can be expanded sequentially to obtain an expression for $p(\mathbf{x}(t_k)|\mathbf{y}_{1:k})$ by induction. We can rewrite Eq. (2.2.6) as

$$p(\mathbf{x}(t_{0:k})|\mathbf{y}_{1:k}) = \prod_{i=2}^{k} \frac{p(\mathbf{y}_i|\mathbf{x}(t_i))p(\mathbf{x}(t_i)|\mathbf{x}(t_{i-1}))}{p(\mathbf{y}_i)} \frac{p(\mathbf{y}_1|\mathbf{x}(t_1))p(\mathbf{x}(t_1)|\mathbf{x}(t_0))p(\mathbf{x}(t_0))}{p(\mathbf{y}_1)} \tag{2.2.7}$$

Integrating both sides of Eq. (2.2.7) with respect to $\mathbf{x}(t_0)$ yields

$$p(\mathbf{x}(t_{1:k})|\mathbf{y}_{1:k}) = \prod_{i=2}^{k} \frac{p(\mathbf{y}_i|\mathbf{x}(t_i))p(\mathbf{x}(t_i)|\mathbf{x}(t_{i-1}))}{p(\mathbf{y}_i)} \underbrace{\frac{p(\mathbf{y}_1|\mathbf{x}(t_1))p(\mathbf{x}(t_1))}{p(\mathbf{y}_1)}}_{=p(\mathbf{x}(t_1)|\mathbf{y}_1)\ by\ Bayes}, \tag{2.2.8}$$

since

$$\int p(\mathbf{x}_{0:k})d\mathbf{x}(t_0) = \prod_{i=2}^{k} p(\mathbf{x}(t_i)|\mathbf{x}(t_{i-1})) \underbrace{\int p(\mathbf{x}(t_1)|\mathbf{x}(t_0))p(\mathbf{x}(t_0))d\mathbf{x}(t_0)}_{p(\mathbf{x}(t_1))} = p(\mathbf{x}(t_{1:k})).$$

Continuing for $i = 2$ we have

$$p(\mathbf{x}(t_{1:k})|\mathbf{y}_{1:k}) = \prod_{i=3}^{k} \frac{p(\mathbf{y}_i|\mathbf{x}(t_i))p(\mathbf{x}(t_i)|\mathbf{x}(t_{i-1}))}{p(\mathbf{y}_i)} \underbrace{\frac{p(\mathbf{y}_2|\mathbf{x}(t_2))p(\mathbf{x}(t_2)|\mathbf{x}(t_1))p(\mathbf{x}(t_1)|\mathbf{y}_1)}{p(\mathbf{y}_2)}}_{p(\mathbf{x}(t_{1:2})|\mathbf{y}_{1:2})}. \quad (2.2.9)$$

Integrating with respect to $\mathbf{x}(t_1)$ in Eq. (2.2.9) yields

$$p(\mathbf{x}(t_{2:k})|\mathbf{y}_{1:k}) = \prod_{i=3}^{k} \frac{p(\mathbf{y}_i|\mathbf{x}(t_i))p(\mathbf{x}(t_i)|\mathbf{x}(t_{i-1}))}{p(\mathbf{y}_i)} \underbrace{\frac{p(\mathbf{y}_2|\mathbf{x}(t_2))p(\mathbf{x}(t_2)|\mathbf{y}_1)}{p(\mathbf{y}_2)}}_{p(\mathbf{x}(t_2)|\mathbf{y}_{1:2})}, \quad (2.2.10)$$

since

$$p(\mathbf{x}(t_2)|\mathbf{y}_1) = \int p(\mathbf{x}(t_2)|\mathbf{x}(t_1))p(\mathbf{x}(t_1)|\mathbf{y}_1)d\mathbf{x}(t_1). \quad (2.2.11)$$

After expanding for the k^{th} instant and integrating sequentially for $\mathbf{x}(t_{k-1})$, we obtain

$$p(\mathbf{x}(t_k)|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}(t_k))p(\mathbf{x}(t_k)|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_{1:k})}, \quad (2.2.12)$$

where

$$p(\mathbf{x}(t_k)|\mathbf{y}_{1:k-1}) = \int p(\mathbf{x}(t_k)|\mathbf{x}(t_{k-1}))p(\mathbf{x}(t_{k-1})|\mathbf{y}_{1:k-1})d\mathbf{x}(t_{k-1}) \quad (2.2.13)$$

Eq. (2.2.12) is the general form of the recursive Bayesian filter. The likelihood function $p(\mathbf{y}_k|\mathbf{x}(t_k))$ represents the pdf of the observations and depends on the noise of the sensor. The posterior pdf before a new observation is incorporated is given by $p(\mathbf{x}(t_k)|\mathbf{y}_{1:k-1})$. Eq. (2.2.13) is known as the Chapman-Kolmogorov equation.

Recursive Estimation Algorithm

After incorporating all the elements that form the Bayesian recursive filter, we have an algorithm that is a recursive process starting with $p(\mathbf{x}(t_0))$, the pdf associated with $\mathbf{x}(t)$ prior to any observations. The recursive algorithm is divided in two main steps, prediction and update, that are applied when each observation $\mathbf{y}_k$ is obtained.

The prediction step is where the pdf prior to an observation, given by $p(\mathbf{x}(t_k)|\mathbf{y}_{1:k-1})$ in Eq. (2.2.13) is calculated. The continuous nature of the system is significant since a stochastic differential equation has to be solved. Theoretically, there are several ways to proceed. Due to the complexity of the models of the system, most of the methods are in general computationally intractable and not suitable for practical applications.

First Method: Propagate the transition density function $p(\mathbf{x}(t_k)|\mathbf{x}(t_{k-1}))$ across the interval $t_{k-1} < t < t_k$ by integrating the stochastic differential equation that represent the state $\mathbf{x}(t)$ from time $t_{k-1} < t < t_k$. Using this result, calculate $p(\mathbf{x}(t_k)|\mathbf{y}_{1:k-1})$ using Eq. (2.2.13). This method is typically computationally intractable, but can be approximated under some assumptions [30].

Second Method: Solve the boundary problem of finding $p(\mathbf{x}(t_k)|\mathbf{y}_{1:k-1})$ starting from the distribution $p(\mathbf{x}(t_{k-1})|\mathbf{y}_{1:k-1})$ and solving the partial differential equation across the interval $t_{k-1} < t < t_k$. It is necessary to use numerical approximations in most cases.

For the update step, we compute the posterior pdf using Bayes theorem to incorporate the observation pdf where,

$$p(\mathbf{x}(t_k)|\mathbf{y_k}) \propto p(\mathbf{y}_k|\mathbf{x}(t_k))p(\mathbf{x}(t_k)|\mathbf{y}_{1:k-1})$$

As mentioned before, this is a general form of the Bayesian estimation. This exact structure will not be readily apparent in most filters, even though, in general, the prediction and update form is followed.

2.3 Motion Models for Tracking on Video

The literature related to pedestrian tracking typically describes a set of simple motion models to model the different possible maneuvers [5]. The most widely used motion models are the constant velocity or linear motion model and the constant acceleration model. For maneuvering targets more realistic models might include the coordinated turn [31] [32] and models with higher orders of dynamic description (for example see [27]) that model the forces acting in the objects to obtain a more detailed representation of the expected motion.

Given the nature of the motion models in the pixel world, it is necessary to describe the relationship between the pixel world and the real 3D world. For this purpose, there is a set of practical tools that allow the change of coordinate systems and projections derived from information about the camera and its position in the scene.

2.3.1 Projective Space

The projective space $\mathbb{P}^n$ of dimension n, is the quotient (the quotient space is the set of equivalence classes) space of $\mathbb{R}^{n+1} \setminus \{0_{n+1}\}$ defined by the following equivalence relation:

$$[x_1, \ldots, x_{n+1}]^t \sim [x'_1, \ldots, x'_{n+1}]^t \Leftrightarrow \exists \lambda \neq 0, [x_1, \ldots, x_{n+1}]^t = \lambda [x'_1, \ldots, x'_{n+1}]^t$$

The points of $\mathbb{P}^n$ that satisfy $x_{n+1} \neq 0$ have an equivalent in the Euclidian space $\mathbb{R}^n$,

$[\frac{x_1}{x_{n+1}}, \ldots, \frac{x_n}{x_{n+1}}]^t$. The points that satisfy $x_{n+1} = 0$ do not have and Euclidean equivalent and are called *the points at infinity*.

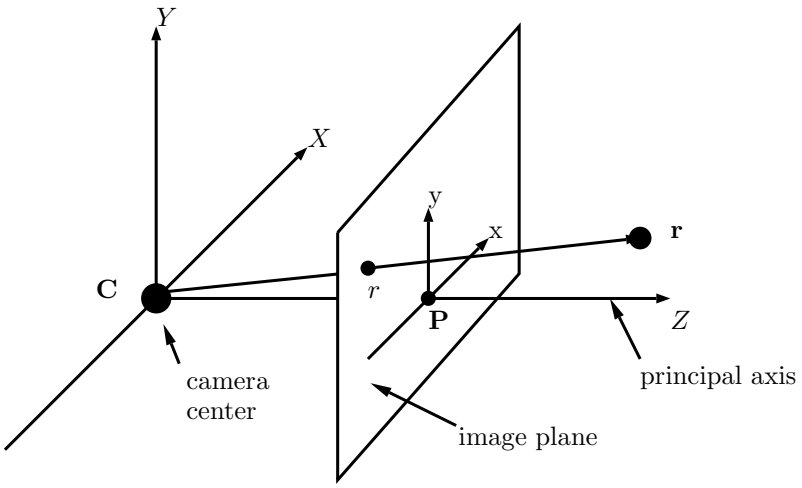


Figure 2.1: Projective space

2.3.2 The Pinhole Camera Model

The pinhole camera model allows us to describe the process of the acquisition of an image by the projection of the 3D points to a set of 2D points situated on the retinal plane. Let C be the optical center of the camera. The projection of a 3D point M is the intersection of the optic ray CM with the retinal plane as you can see in Figure 2.2.

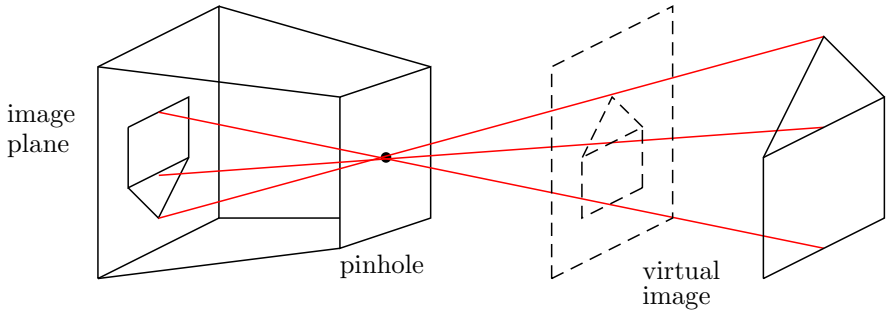


Figure 2.2: The pinhole camera model

Let $M = [x\ y\ z]^t$ be a point of the Euclidean 3D space and $m = [u\ v]^t$ its projection.

Let $\widetilde{M} = [x\ y\ z\ 1]^t$ and $\widetilde{m} = [u\ v\ 1]$ be the corresponding points in homogeneous coordinates. In order to formalize the projection process we write:

$$\mathcal{P}\widetilde{M} = \widetilde{m}$$

where $\mathcal{P}$ is a 3×4 projection matrix.

2.3.3 Decomposition of the Projection Matrix

In this part we define two coordinate frames: one linked to the scene (which we can choose) and the other linked to the camera. The origin of the camera reference frame is the center of the camera and its axes are the axes of the retinal plane and the optical axis (this is the axis that passes from the camera center and is normal to the retinal plane). This can be seen in Figure 2.1.

It can be shown that the projection matrix $\mathcal{P}$ is decomposed as follows:

$$\mathcal{P} = \mathbf{K}\,[\mathbf{R}|\mathbf{T}]$$

where: $\mathbf{K}$ is a 3×3 matrix that contains the *intrinsic* parameters of the camera (these are the parameters that depend only on the internal configuration of the camera). This matrix describes the reference frame of the retinal plane. We can write it

$$\mathbf{K} = \begin{bmatrix} f_u & \gamma & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

where f_u and f_v are the focal distances expressed in pixels. $[u_0\ v_0]$ are the coordinates of the principal point, i.e. the intersection point of the retinal plane with the optical axis. Finally γ is called *skew* parameter and is 0 for most normal cameras. The matrix $[\mathbf{R}|\mathbf{T}]$ represents the

pose of the camera in the scene reference frame. This rigid transformation allows us to transform the 3D points of the scene reference frame to the camera reference frame. The matrix $\mathbf{R} \in \Re^{3\times3}$ rotation matrix and $\mathbf{T} \in \Re^{3\times1}$ is a translation vector. The quantities $\mathbf{R}$ and $\mathbf{T}$ are the *extrinsic* parameters of the camera.

Note that the projection matrix depends on 11 parameters: 5 intrinsic parameters and 6 extrinsic parameters (3 for rotation and 3 for translation). The process of *calibration* of a camera consists of estimating the intrinsic and/or extrinsic parameters.

2.4 Pedestrian Measurement Models

It is important to note that an exact measurement model is not possible in practice since the way in which observations are obtained on videos since these observations are the result of image processing techniques applied to each frame. Depending on the application there are several ways in which observations can be obtained given that feature extraction and detection is a field of research by itself that applies classical signal processing techniques, such as digital filters, or machine learning techniques such as support vector machines and convolutional neural networks, to obtain useful information from images. In the case concerning estimation and tracking we are only concerned with the output given by these techniques.

The output of an object detector (example of usual objects are faces, pedestrians, cars) usually gives a bounding box that can be used on tracking schemes that are being performed on the pixel world. From this then we can obtain direct measurement of the centroid of the object (or objects) being tracked and also its relative size in the image, in the case of pedestrians for example we have a similar visual output to the one shown in Figure

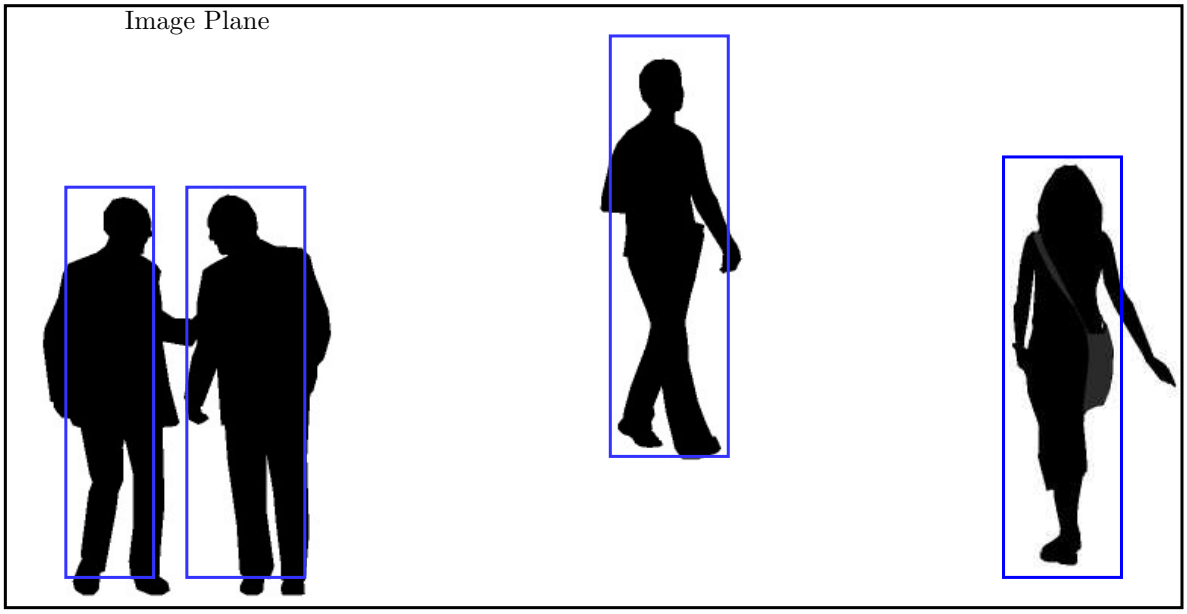


Figure 2.3: Measurement visualization

2.3. For a tracking application where the centroid position and speed are being estimated Eq. (2.2.2) can represented by a linear vector equation on the pixel world

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} \tag{2.4.1}$$

with

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{x} = \begin{bmatrix} r_x \\ \dot{r}_x \\ r_y \\ \dot{r}_y \end{bmatrix},$$

this can be transformed to world coordinates (if necessary) using the matrices and techniques described earlier in this chapter.

CHAPTER 3

Multi-target Tracking Techniques

This chapter presents general representations of multiple target tracking (MTT) techniques from the general Bayesian representation point of view. This subject is introduced with the aim of relating the techniques that are studied in this work with the classical estimation framework that was introduced in Chapter 2, but extended for the sequential MTT problem. The discussion here is focused on describing a concise set of the solutions coming from the Bayesian and probabilistic point of view that apply further modification and assumptions to the procedure shown in general representation. Since there are a wide variety of MTT techniques that take a Bayesian approach [2], describing the mathematical details and assumptions is beyond the scope of this work. We focus on four representative solutions: Multiple Hypothesis Tracking, Joint Probabilistic Data Association, Markov Chain Monte Carlo for tracking and Probability Hypothesis Density filter. These techniques are describe throughout the next sections and the rest of the chapter but are not the only ones in existence, more details of the techniques described here and many more can be found in [2][32][33][34][35]. It is important to note that the presentation for the Multiple Hypothesis Tracker and Joint Probabilistic Data Association borrows elements and structure of the one given in [35].

3.1 General Bayesian Formulation

Let us start by introducing a really general representation of multiple target state spaces and measurement spaces. In chapter 2 single target state space models were introduced and represented by Eq. 2.2.1, now we can generalize that definition for multiple targets (focusing on a sequential point of view, important for real time implementations) assuming that they have the same representation and also that a given instant of time can be present (or exist) or not. Then we can assume a general state space represented by $\mathcal{S}$ where all target $\boldsymbol{x}(t)$ move. At any given instant of time the total number of target $\bar{N}(t)$ is unknown. We can designate a region, $\mathcal{R}$, which defines the boundary of the tracking problem. Given this region we can then add an additional state ϕ to the target state space $\mathcal{S}$ that denotes if a target is not inside the defined boundary $\mathcal{R}$, then $\mathcal{S}^+ = \mathcal{S} \cup \{\phi\}$ and this is true for each target, giving us a joint state space $\mathcal{S} = \mathcal{S}^+ \times \cdots \times \mathcal{S}^+$ where the product is taken $\bar{N}(t)$. With all this in mind we can represent the set of target at any given time $t > 0$ taking in account that new targets can be born

$$\boldsymbol{X}(t) = \{\boldsymbol{x}_1(t), \boldsymbol{x}_2(t), ..., \boldsymbol{x}_{n'}(t)\} \cup \{\boldsymbol{b}_1, ..., \boldsymbol{b}_\nu\} \tag{3.1.1}$$

where $\boldsymbol{x}(t)_{1...n'}$ are the target that persist from the last instant of time and $\boldsymbol{b}_{1...\nu}$ are the new "born" targets.

For the measurement representation we have the representation from Eq. 2.2.2. We can extend this to a set of $\bar{M}$ measurements that in general can be produced from the targets in the set $\boldsymbol{X}$ or by false alarms (clutter in the environment) or wrong measurements. This set can be represented as

$$\boldsymbol{Y}_k = \{\boldsymbol{y}_{1,k}, ..., \boldsymbol{y}_{\bar{M},k}\} \tag{3.1.2}$$

The objective of multiple target Bayesian estimation in this case is to estimate the contents of the set $\boldsymbol{X}(t)$ recursively, based on the set of observations $\boldsymbol{Y}_k$, using the joint transition density for the state $p(\boldsymbol{X}(t_k)|\boldsymbol{X}(t_{k-1}))$ and the joint likelihood function $p(\boldsymbol{Y}_k|\boldsymbol{X}(t_k))$. We have also the assumptions that are key to Bayesian estimation and inference described for MTT. First, the Markov assumption which states that the values in any set of states $\boldsymbol{X}(t)$ are only influenced by the values of the set of states $\boldsymbol{X}(t-dt)$ that directly preceded it. This implies that the past is independent of the future. In a continuous-discrete setting, we have

$$p(\boldsymbol{X}(t_{0:k})) = \prod_{i=1}^{k} p(\boldsymbol{X}(t_i)|\boldsymbol{X}(t_{i-1}))p(\boldsymbol{X}(t_0)). \tag{3.1.3}$$

We also have the conditional independence of the set of observations that states that the observation set, $\boldsymbol{Y}_k$, given the state, $\boldsymbol{X}(t_k)$, is conditionally independent from the observation and state history, or

$$\begin{aligned} p(\boldsymbol{Y}_{1:k}) &= \prod_{i=1}^{k} p(\boldsymbol{Y}_i) \\ p(\boldsymbol{Y}_{1:k}|\boldsymbol{X}(t_{0:k})) &= \prod_{i=1}^{k} p(\boldsymbol{Y}_i|\boldsymbol{X}(t_i)). \end{aligned} \tag{3.1.4}$$

Finally the estimation process ideally follows the same procedure as the single target but with the use of the joint densities given the state at time step t_{k-1}, Bayes theorem is used to determine the joint posterior density at time t_k. It is achieved in two steps: Given

the motion model and the Bayesian joint posterior density $p(\boldsymbol{X}(t_{k-1})|\boldsymbol{Y}_{k-1})$ at time t_{k-1}, a time-updated joint density is obtained using the Chapman-Kolmogorov equation:

$$p(\boldsymbol{X}(t_k)|\boldsymbol{Y}_{k-1}) = \int p(\boldsymbol{X}(t_k)|\boldsymbol{X}(t_{k-1}))p(\boldsymbol{X}(t_{k-1})|\boldsymbol{Y}_{k-1})\mathrm{d}\boldsymbol{X}(t_{k-1}) \tag{3.1.5}$$

The observation set $\boldsymbol{Y}_k$ is used to update (weight) the density produced by the time-update step to determine the final joint posterior density at time t_k:

$$p(\boldsymbol{X}(t_k)|\boldsymbol{Y}_k) = \frac{p(\boldsymbol{Y}_k|\boldsymbol{X}(t_k))p(\boldsymbol{X}(t_k)|\boldsymbol{Y}_{1:k-1})}{p(\boldsymbol{Y}_{1:k})}, \tag{3.1.6}$$

The joint posterior density function $p(\boldsymbol{X}(t_k)|\boldsymbol{Y}_k)$ encapsulates everything about the set of target states, based on the current set of observations and a priori information. The calculations needed to obtain the exact posterior of this unified estimation are even more unrealistic than in the single object case and the different algorithms that have been designed to do it need some extra assumptions that facilitate implementations of sequential solutions.

3.1.1 Assumptions for Classical MTT techniques

In general there is a group of assumptions that change in between techniques and define the approach taken to solve the MTT problem:

DA.1 A measurement can originate from at most one target or from clutter.

DA.2 A target can generate at most one measurement at every time step.

DA.1a A measurement can originate from one target, multiple targets or from clutter.

DA.2a A target can generate zero or multiple measurements at every time step.

These assumptions are the way in which different algorithms manage the problem of data association between measurements and targets (defined in chapter 1). Also target

identity in the sense that targets can be superposed or not to manage occlusions (represented on assumption **DA.1a**), but not in the identity of targets (labeling) from instant of time to instant of time. Labeling is usually an added feature to each technique but it is not part of the classical implementation, at least in their probabilistic formulation. These assumptions will be useful to describe the different techniques and their relationships.

3.2 Non-Strictly Bayesian Solutions to the MTT problem

There have been many techniques used in the literature to solve the MTT problem in a recursive way that can be based on heuristics or probabilistic assumption and auxiliary structures. For example a general filter that was proposed and has been widely used is the Nearest neighbor standard filter [33], that has many variants including probabilistic nearest neighbor, suboptimal nearest neighbor, and global nearest neighbor (GNN) [32]. The main idea is to use a validation gate to find out if the measurements correspond to the right target and in some occasions to test for track survival. This gate usually has a Gaussian shape with the covariance given by the innovation of the measurements (difference between measurements, $\boldsymbol{y}$, and predicted measurements $\hat{\boldsymbol{y}}$), all the variations depend on the way the validation is performed. This gating process is also used as a measurement preprocessing for other MTT techniques [36] [32] to have the measurements ready for the data association assumptions (see figure 3.1). These algorithms usually only have a limited performance restricted to widely spaced target and rely heavily on extra heuristics [34].

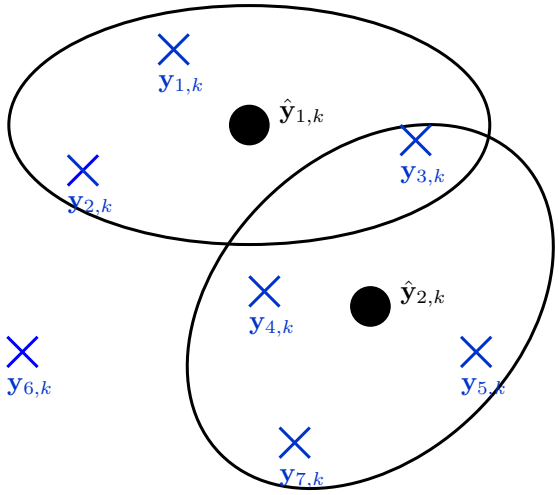


Figure 3.1: 2D example of gating (Nearest Neighbor approach) for measurement and data association

3.3 Multiple Hypotheses Tracker

The Multi Hypothesis Tracker (MHT) is a method for calculating the probabilities of various data association hypothesis. It maintains several hypotheses for each target at each instant of time. In order to do this, this techniques enumerates all possible associations over time. As each measurement is obtained, it is classified according to its probability of origin: coming from a previously known target, or from a false measurement, or from a new target. The estimation of each possible hypothesis is done through the Kalman filter (for Gaussian transition densities as introduced in the original literature [37]). With the gathering of more information (or measurements), the probabilities of joint hypotheses are calculated sequentially using all the supposed known information about the system such as density of unknown target, probability of detection, and density of false targets, this general technique is usually regarded as a hypotheses oriented or Measurement-to-target ($M \rightarrow T$) data association [35].

This evaluation process then is a deterministic and exhaustive way of enumerating all the possible associations, this branching techniques allows correlation of a measurement with

its source based on past, present and future data. This makes this technique computationally exponential in both time and memory. To keep the number of hypotheses reasonable and the algorithm computationally tractable, a number of techniques have been developed to eliminate unlikely hypotheses and to combine hypotheses with similar target estimates. The most common techniques are clustering (part of the original work by Reid [38]) and k-best hypotheses generation (the most widely used based on Murty's work [39]) [32].

The general process of the algorithm can be then divided in 4 steps (two more than the unified and theoretical tracking described in Eq. 3.1.6) including all the association processes. First as we mentioned, it works as a branching process meaning that it recursively maintains a hypothesis tree that expands each instant of time with a new set of hypotheses:

1. All the hypotheses on the latest leaf (or initial leaf for $k = 1$) are predicted according to their transition model (Common *Prediction* step).

2. New hypotheses are generated from the leaf hypotheses of the hypothesis tree (*hypothesis generation*).

3. The probabilities of the new hypotheses are calculated (*hypothesis evaluation*) and the ones with low probability are discarded (*pruning*).

4. The surviving hypotheses are updated according to the data association of the new hypothesis (Common *Update* step).

In order to introduce a formal presentation of the process described above we should introduce a new set to represent the hypothesis and probability of association. For the hypothesis set we have

$$\mathbf{\Lambda}_{t_k}^h = \left\{ \lambda_{t_i}^h \right\}_{t_i=1:k} \tag{3.3.1}$$

this represents a hypothesis h at time step t_k, that is formed by the set of association events $\lambda_{t_k}^h$ up to time step t_k. The association events $\lambda_{t_k}^h$ are finally formed by measurement to target associations indicating which measurement originate from which states at time step t_k,

$$\lambda_{t_k}^h = \left\{ \boldsymbol{r}_{t_k}^q \right\}^{q=1:\bar{Q}(t_k)}. \tag{3.3.2}$$

The association event $\boldsymbol{r}_{t_k}^q$ in this case represents the measurement-to-target association and it is a discrete variable taking a value between 0 and $\bar{N}(t)$ indicating if the measurement $\boldsymbol{y}_k$ is a false measurement or by which target it is caused.

The algorithm also maintains a set of target trackers for each state being estimated (Equation 3.1.1) but restricted to the hypothesis leafs, and alone side the hypothesis set $\mathbf{\Lambda}_{t_k}^h$. Given this, we can represent the state as $\boldsymbol{x}_n^h(t_k)$ of the $\bar{N}^h(t_k)$ targets in the hypothesis. To further simplify the process, target dynamics are assumed independent, then the target states in a hypothesis h can be simplified as a product of independent probability density functions (where $p(h)$ represents the parent hypothesis)

$$p(\boldsymbol{X}^h(t_k)|\boldsymbol{X}^{p(h)}(t_{k-1})) = \prod_{n=1}^{\bar{N}^h(t)} p(\boldsymbol{x}^h(t_k)|\boldsymbol{x}^{p(h)}(t_{k-1})), \tag{3.3.3}$$

and then the prediction can be done independently for each target using Eq. (2.2.13) (Chapman-Kolmogorov equation).

The process of *hypothesis generation* consist of creating a new set of hypotheses $\left\{\mathbf{\Lambda}_{t_k}^h\right\}^{h=1:\bar{H}(t_k)}$ from the previous time step by combining one hypothesis at time step t_{k-1} with an association event $\lambda_{t_k}^h$

$$\mathbf{\Lambda}_{t_k}^h = \left\{\mathbf{\Lambda}_{t_{k-1}}^{p(h)}, \lambda_{t_k}^h\right\} \tag{3.3.4}$$

The association events represented in Eq. (3.3.2) consist on finding out which measurement originate from clutter, from which target or from new targets (birthed or spawned targets), this requires the enumeration of all possible association events that are limited by the assumptions in section 3.1.1, more specifically for the classical implementation of the MHT assumption **DA**.**1** and **DA**.**2**.

To rank the different hypotheses or more specifically to do *hypothesis evaluation*, it is necessary to define a weighting factor for each hypothesis. The importance of the hypothesis $\mathbf{\Lambda}_{t_k}^h$ is expressed by the conditional probability density function $p(\mathbf{\Lambda}_{t_k}^h|\boldsymbol{y}_{1:k})$. Taking in account the definition of the hypothesis set and the acquisition of a new association 3.3.4 we can use Bayes rule to rewrite $p(\mathbf{\Lambda}_{t_k}^h|\boldsymbol{y}_{1:k})$ as:

$$\begin{aligned} p(\mathbf{\Lambda}_{t_k}^h|\boldsymbol{y}_{1:k}) &\propto p(\boldsymbol{y}_k|\mathbf{\Lambda}_{t_k}^h, \boldsymbol{y}_{1:k-1})p(\mathbf{\Lambda}_{t_k}^h|\boldsymbol{y}_{1:k-1}) \\ &\propto p(\boldsymbol{y}_k|\mathbf{\Lambda}_{t_k}^h, \boldsymbol{y}_{1:k-1})p(\lambda_{t_k}^h, \mathbf{\Lambda}_{t_{k-1}}^{p(h)}|\boldsymbol{y}_{1:k-1}) \\ &\propto p(\boldsymbol{y}_k|\mathbf{\Lambda}_{t_k}^h, \boldsymbol{y}_{1:k-1})p(\lambda_{t_k}^h|\mathbf{\Lambda}_{t_{k-1}}^{p(h)}, \boldsymbol{y}_{1:k-1})p(\mathbf{\Lambda}_{t_{k-1}}^{p(h)}|\boldsymbol{y}_{1:k-1}) \end{aligned} \tag{3.3.5}$$

Each of the probability density in Eq. (3.3.5) has an interpretation:

· The pdf $p(\boldsymbol{y}_k|\mathbf{\Lambda}_{t_k}^h, \boldsymbol{y}_{1:k-1})$ represents the probability of obtaining a group of

measurements given all past associations. Here is where the distinction between measurements originated from existing targets, from new targets or clutter is made. The association event $\lambda_{t_k}^h$ contains a set of F_k false elements and set of ν_k measurement from new targets and the remaining measurements $\bar{Q}(t_k) - F_k - \nu_k$ come from existing targets. Using the conditional independence of observations from Equation 2.2.4

$$p(\boldsymbol{y}_k|\boldsymbol{\Lambda}_{t_k}^h, \boldsymbol{y}_{1:k-1}) = \prod_{q:\boldsymbol{r}_{t_k}^q=0} p_C \prod_{q:\boldsymbol{r}_{t_k}^q>\bar{N}^h(t_k)} p_B \prod_{q:\boldsymbol{r}_{t_k}^q\in\left\{1:\bar{N}^h(t_{k-1})\right\}} p(\boldsymbol{y}_k^q|\boldsymbol{\Lambda}_{t_k}^h, \boldsymbol{y}_{1:k-1}) \quad (3.3.6)$$

The first factor represents the false alarms and clutter measurement (with pdf p_C), the second factor represent the birth of new targets (with pdf p_B) and finally the third term represents the existing targets. The pdf $p(\boldsymbol{y}_k^q|\boldsymbol{\Lambda}_{t_k}^h, \boldsymbol{y}_{1:k-1})$ is required to calculate Eq. (3.3.6), this represent the probability that a measurement is originated from a target given all the measurements up to time step t_{k-1}. This pdf can be de-marginalized and represented in term of the association events and the known densities (or supposed known), then we can rewrite it as

$$\begin{aligned}
p(\boldsymbol{y}_k^q|\boldsymbol{\Lambda}_{t_k}^h, \boldsymbol{y}_{1:k-1}) &= \int_{\boldsymbol{x}^{\boldsymbol{r}_{t_k}^q(t_k)}} p(\boldsymbol{y}_k^q, \boldsymbol{x}^{\boldsymbol{r}_{t_k}^q}(t_k)|\boldsymbol{\Lambda}_{t_k}^h, \boldsymbol{y}_{1:k-1}) \mathrm{d}\boldsymbol{x}^{\boldsymbol{r}_{t_k}^q}(t_k) \\
&= \int_{\boldsymbol{x}^{\boldsymbol{r}_{t_k}^q(t_k)}} p(\boldsymbol{y}_k^q|\boldsymbol{x}^{\boldsymbol{r}_{t_k}^q}(t_k), \boldsymbol{\Lambda}_{t_k}^h, \boldsymbol{y}_{1:k-1}) p(\boldsymbol{x}^{\boldsymbol{r}_{t_k}^q}(t_k)|\boldsymbol{\Lambda}_{t_k}^h, \boldsymbol{y}_{1:k-1}) \mathrm{d}\boldsymbol{x}^{\boldsymbol{r}_{t_k}^q}(t_k) \\
&= \int_{\boldsymbol{x}^{\boldsymbol{r}_{t_k}^q(t_k)}} p(\boldsymbol{y}_k^q|\boldsymbol{x}^{\boldsymbol{r}_{t_k}^q}(t_k)) p(\boldsymbol{x}^{\boldsymbol{r}_{t_k}^q}(t_k)|\boldsymbol{y}_{1:k-1}) \mathrm{d}\boldsymbol{x}^{\boldsymbol{r}_{t_k}^q}(t_k) \qquad (3.3.7)
\end{aligned}$$

where $p(\boldsymbol{y}_k^q|\boldsymbol{x}^{\boldsymbol{r}_{t_k}^q}(t_k))$ is the observation model and $p(\boldsymbol{x}^{\boldsymbol{r}_{t_k}^q}(t_k)|\boldsymbol{y}_{1:k-1})$ is the density obtained from the *Prediction* step conditioned to the discrete association vector $\boldsymbol{r}_{t_k}^q$

from the last instant of time.

- The pdf $p(\lambda_{t_k}^h|\mathbf{\Lambda}_{t_{k-1}}^{p(h)}, \boldsymbol{y}_{1:k-1})$ represents the probability of an association event given all the past measurements and data associations. Here is again the pdf of false alarms, new measurements and track detection are taken into account. According to the MHT literature [38] [34] [33] it can be represented as

$$\begin{aligned} p(\lambda_{t_k}^h|\mathbf{\Lambda}_{t_{k-1}}^{p(h)}, \boldsymbol{y}_{1:k-1}) &= \frac{\nu_k! F_k!}{\bar{Q}(t_k)!} P(\nu) P(F) \\ & \prod_{n=1}^{\bar{N}(t)} (P_k^{det,n})^{1_k^{det,n}} (1 - P_k^{det,n})^{1-1_k^{det,n}} \\ & (P_k^{del,n})^{1_k^{del,n}} (1 - P_k^{del,n})^{1-1_k^{del,n}} \end{aligned} \tag{3.3.8}$$

where $P(\nu)$ is the prior for the target birth density, $P(F)$ is the prior for false measurements (clutter), both usually assumed to be poisson or uniformly distributed, and in general 3.3.8 has a multinomial pdf form taking in account the probability of detection $P_k^{det,n}$ and deletion $P_k^{del,n}$ (using their respective indicator functions) and are assumed to be constant.

- The pdf $p(\mathbf{\Lambda}_{t_{k-1}}^{p(h)}|\boldsymbol{y}_{1:k-1})$ simply represents the probability of the parent hypothesis given all the previous measurements and it is available recursively (from the previous instant of time).

After obtaining all the probabilities for the different hypothesis using the set of equations formerly described, *pruning* is done by dismissing the hypotheses that obtained low values, this in order to maintain the tractability of the process.

Finally, the *Update* simply consists on performing the basic Bayesian correction Eq. (2.2.12) (approximated) to all the targets that survive the prunning, according to the

measurement association event

$$p(\boldsymbol{x}^{h,n}(t_k)|\boldsymbol{y}_{1:k}) \propto p(\boldsymbol{y}_k^q|\boldsymbol{x}^{h,n}(t_k))p(\boldsymbol{x}^{h,n}(t_k)|\boldsymbol{y}_{1:k-1}) \quad \text{with} \quad q : \boldsymbol{r}_{t_k}^q \in \lambda_{t_k}^h = n \tag{3.3.9}$$

which uses the measurement model and the target prediction in the usual manner.

3.3.1 Gaussian Approximation

The MHT algorithm just described can be simplified by approximating the posterior of the joint target state, transition and likelihood models with Gaussian distributions (we are going to the same assumption through out this work to facilitate implementation and mathematical treatments). The approximation of the joint target state $p(\boldsymbol{X}(t_k)|\boldsymbol{Y}_k)$ is done by a factorial form where each factor is the marginal distribution $p(\boldsymbol{x}^n(t_k)|\boldsymbol{y}_{1:k})$ corresponding to a single target, then obtaining the representation

$$p(\boldsymbol{X}(t_k)|\boldsymbol{Y}_k) \approx \prod_{n=1}^{\bar{N}(t)} p(\boldsymbol{x}^n(t_k)|\boldsymbol{y}_{1:k}) \approx \prod_{n=1}^{\bar{N}(t)} \mathcal{N}(\boldsymbol{x}^n(t_k)|\boldsymbol{\mu}^n(t_k), \boldsymbol{\Sigma}^n) \tag{3.3.10}$$

where $\boldsymbol{\mu}^n(t_k)$ and $\boldsymbol{\Sigma}^n$ are mean and covariance of the n^{th} target. The Gaussian assumption of the transition and likelihood models reduces the *prediction* and *udpate* step to a regular Kalman filter acting on each of the targets separately.

One last expression that represents the *hypothesis evaluation* in a simple fashion comes naturally from the Gaussian assumption and from the assumption that new targets and false measurements are uniformly distributed over the observation volume V, changing Eq. (3.3.6) to

$$p(\boldsymbol{y}_k|\boldsymbol{\Lambda}^h_{t_k}, \boldsymbol{y}_{1:k-1}) = V^{-(F_k+\nu_k)} \prod_{q:\boldsymbol{r}^q_k \in \left\{1:\bar{N}^h(t_{k-1})\right\}} \mathcal{N}(\boldsymbol{y}^q_k|\bar{\boldsymbol{y}}^{\boldsymbol{r}^q_k}_k, \boldsymbol{S}^{\boldsymbol{r}^q_k}) \tag{3.3.11}$$

where $\bar{\boldsymbol{y}}^{\boldsymbol{r}^q_k}_k$ is the predicted mean (using the measurement model) and $\boldsymbol{S}^{\boldsymbol{r}^q_k}$ is the innovation covariance (as calculated for the standard Kalman filter), all of this dependent of the previous existence as determined by the association event $\boldsymbol{r}^q_k$.

The implementation used in this work comes from the libraries presented by Antunes et al. [40] mainly using java but that can be called from Matlab. It can be obtained from `multiplehypothesis.com`.

3.4 Joint Probabilistic Data Association

The Joint Probabilistic Data Association is an extension of the PDA method, which dealt with a single target in clutter, to the situation where there is a *known* number of targets in clutter. When there are several targets in the same neighborhood, measurements from one target can fall in the validation region of a neighboring target. This can happen over several sampling times and acts as "persistent interference". Since the PDA algorithm models all the incorrect measurements as "random interference", with independent uniform spatial distributions, its performance can degrade significantly when the existence of a neighboring target gives rise to interference that is not correctly modeled.

The probabilistic data association (PDA) approach avoids ambiguous decisions by averaging over the different data association hypotheses. The PDA approach was originally developed for tracking a single target under clutter [33] but was modified for application in multitarget environments, referred to as the joint probabilistic data association (JPDA) [41].

The JPDA algorithm is the best known example of the Bayesian data association paradigm.

The JPDA tackles uncertain data association conditions by allowing a target to be updated by a weighted sum of all measurements (in its gate). The weights represent the probability that the measurement originates from that target. As such, a measurement can contribute to more than one track, and it contributes to this target with a certain weight. Such associations are referred to as soft assignments as compared to the hard assignments of the NN and MHT algorithms. In order to determine these weights, the probability that each measurement originates from each target has to be calculated. To this end, in the most basic setting, all possible hypotheses have to be enumerated at every time step. To construct these hypotheses the assumptions DA1a and DA2a are made: a measurement can only originate from a single target and a target cannot generate more than one measurement. To limit the number of hypotheses, Murtys' algorithm [39] can be used, as in the MHT, to only generate hypotheses with considerable probabilities, by determining the k-best hypotheses in polynomial time.

The assumption for the transition densities for the targets and target prediction for the JPDA algorithm have a similar nature to the MHT, assuming independence in both cases, but even simpler given that the number of targets is set and there is no need to incorporate target death and birth. The difference and the problem for the JPDA arises during the *update* stage, where usually we would have the equation

$$p(\boldsymbol{x}(t_k)|\boldsymbol{y}_{1:k}) \propto p(\boldsymbol{y}_k|\boldsymbol{x}(t_k))p(\boldsymbol{x}(t_k)|\boldsymbol{y}_{1:k-1}) \tag{3.4.1}$$

but this step cannot be performed independently for each target. The JPDA circumvents

this problem by using a strategy where the data association uncertainty is solved by updating each target's state with a weighted set of measurements. These weights represent an approximation of the posterior probability of the measurements coming from that target. This technique is called a *soft* assignment and is based on a redefinition of the likelihood of each target as a mixture

$$p(\boldsymbol{y}_k|\boldsymbol{x}(t_k)) = \sum_{q=1}^{Q} c_{t_k}^{q,n} p(\boldsymbol{y}_{1:k}|\boldsymbol{x}^n(t_k)) \tag{3.4.2}$$

with $c_{t_k}^{q,n} = p(\boldsymbol{r}_{t_k}^q = n|\boldsymbol{y}_{1:k})$ meaning the posterior probability that measurement $\boldsymbol{y}_k$ is caused by target n. The JPDA treats the data as joint association events that contain the association of measurement to target, represented as $\lambda_{t_k} = \left\{\boldsymbol{r}_{t_k}^q\right\}^{q=1:\bar{Q}(t_k)}$ (similar to the MHT without the hypothesis consideration). Then the final representation for $c_{t_k}^{q,n}$ can be given by

$$c_{t_k}^{q,n} = \sum_{\lambda_{t_k} \in \boldsymbol{\Lambda}_{t_k}^{q,n}} p(\lambda_{t_k}|\boldsymbol{y}_{1:k}) \tag{3.4.3}$$

with $\boldsymbol{\Lambda}_{t_k}^{q,n}$ representing the joint association events that assign measurements to target, $\boldsymbol{\Lambda}_{t_k}^{q,n} = \left\{\lambda_{t_k} : \boldsymbol{r}_{t_k}^q = n \in \lambda_{t_k}\right\}$. In order to calculate the pdf in Eq. (3.4.3) we can use the Markov assumption and Bayes theorem to obtain a marginalized representation that can be numerically evaluated

$$
\begin{aligned}
p(\lambda_{t_k}|\boldsymbol{y}_{1:k}) &\approx \int p(\lambda_{t_k}|\boldsymbol{x}(t_k),\boldsymbol{y}_k)p(\boldsymbol{x}(t_k)|\boldsymbol{y}_{1:k-1})d\boldsymbol{x}(t_k) \\
&\propto \int p(\lambda_{t_k}|\boldsymbol{x}(t_k))p(\boldsymbol{y}_k|\lambda_{t_k},\boldsymbol{x}(t_k))p(\boldsymbol{x}(t_k)|\boldsymbol{y}_{1:k-1})d\boldsymbol{x}(t_k) \qquad (3.4.4)
\end{aligned}
$$

Each of the probability densities in Eq. (3.4.4) has an interpretation:

· The pdf $p(\lambda_{t_k}|\boldsymbol{x}(t_k))$ represents the probability of association event λ_{t_k} given the current state. The most common assumptions for this pdf are that all the assignments have the same likelihood, meaning $p(\lambda_{t_k}|\boldsymbol{x}(t_k)) = constante$ [34], or that the clutter in the environment is taken into account (false measurements) [35]. This yields

$$
p(\lambda^h_{t_k}|\boldsymbol{\Lambda}^{p(h)}_{t_{k-1}},\boldsymbol{y}_{1:k-1}) = \frac{F_k!}{\bar{Q}(t_k)!}P(F)\prod_{n=1}^{N(t)}(P_k^{det,n})^{1^n_k}(1-P_k^{det,n})^{1-1^n_k} \qquad (3.4.5)
$$

where $P(F)$ is the prior for false measurements (clutter) usually assumed to be Poisson (parametric JPDA) or uniformly distributed (non-parametric JPDA).

· The pdf $p(\boldsymbol{y}_k|\lambda_{t_k},\boldsymbol{x}(t_k))$ denotes the probability of obtaining the measurement $\boldsymbol{y}_k$ given the states $\boldsymbol{x}(t_k)$ and the association events λ_{t_k}. Here we make once again the assumption that measurement originated from clutter and targets are independent, having a set of F false alarms and set of $Q(t) - F$ measurements, each coming from a single target. Under this assumption, we see

$$
p(\boldsymbol{y}_k|\boldsymbol{\Lambda}_{t_k},\boldsymbol{y}_{1:k-1}) = \prod_{q:\boldsymbol{r}_{t_k}=0} p_C \prod_{q:\boldsymbol{r}^q_{t_k}\in\{1:N(t_k)\}} p(\boldsymbol{y}^q_k|\boldsymbol{x}^{\boldsymbol{r}^q_{t_k}}(t_k)) \qquad (3.4.6)
$$

where p_C is the pdf of clutter, usually assumed uniform in the observation volume.

· The last pdf is simply the result of the prediction step done to each target independently.

Having all the necessary representations for the pdfs in Eq. (3.4.4) and adding the assumption that clutter probability in Eq. (3.4.6) has the mentioned uniform representation, we can replace all the factors in Eq. (3.4.3), obtaining

$$c_{t_k}^{q,n} \propto \sum_{\lambda_{t_k} \in \mathbf{\Lambda}_{t_k}^{q,n}} \left[V^{-F_k} \frac{F_k!}{\bar{Q}(t_k)!} P(F) \prod_{n=1}^{N(t)} (P_k^{det,n})^{1_k^n} (1 - P_k^{det,n})^{1-1_k^n} \right.$$
$$\left. \prod_{q:\boldsymbol{r}_{t_k}^q \in \{1:N(t_k)\}} \int p(\boldsymbol{y}_k^q | \boldsymbol{x}^{\boldsymbol{r}_{t_k}^q}(t_k)) p(\boldsymbol{x}^{\boldsymbol{r}_{t_k}^q}(t_k) | \boldsymbol{y}_{1:k-1}) d\boldsymbol{x}^{\boldsymbol{r}_{t_k}^q}(t_k) \right]. \quad (3.4.7)$$

The integral in the last part of this expression is simply the predicted likelihood and its computation will result on the on innovation pdf. Evaluating Eq. (3.4.7) would be intractable for a large number of measurements and targets given that it is necessary to enumerate all possible association events assigning measurements to target. This can be limited using a similar approach than the MHT (Murthy's algorithm [42]) or gating [34]. Other important things to take in account for implementation purposes of this algorithm are: all the joint association events are assumed independent over time, and it is assumed that each measurement has the same probability of coming from any target.

3.4.1 Gaussian Approximation

The JPDA algorithm just described can be simplified by approximating the posterior of the joint target state, transition and likelihood models with Gaussian distributions (we are going to use the same assumption through out this work to facilitate implementation and

mathematical treatments). The approximation of the joint target state $p(\boldsymbol{X}(t_k)|\boldsymbol{Y}_k)$ is done by a factorial form where each factor is the marginal distribution $p(\boldsymbol{x}^n(t_k)|\boldsymbol{y}_{1:k})$ corresponding to a single target, then obtaining the representation

$$p(\boldsymbol{x}^n(t_k)|\boldsymbol{y}_{1:k}) \approx \mathcal{N}(\boldsymbol{x}^n(t_k)|\boldsymbol{\mu}^n(t_k), \boldsymbol{\Sigma}^n) \tag{3.4.8}$$

where $\boldsymbol{\mu}^n(t_k)$ and $\boldsymbol{\Sigma}^n$ are mean and covariance of the target n. The Gaussian assumption of the transition and likelihood models reduces the *prediction* step to a regular Kalman filter acting on each of the targets separately.

With this technique the difference comes at the moment of doing the *update* since it is necessary to find the weights for each measurement to target before we can perform it. In order to evaluate this, Eq. (3.4.7) can changed by

$$c_{t_k}^{q,n} \propto \sum_{\lambda_{t_k} \in \boldsymbol{\Lambda}_{t_k}^{q,n}} \left[V^{-F_k} \frac{F_k!}{\bar{Q}(t_k)!} P(F) \prod_{n=1}^{N(t)} (P_k^{det,n})^{1_k^n} (1 - P_k^{det,n})^{1-1_k^n} \prod_{q:\boldsymbol{r}_{t_k}^q \in \{1:N(t_k)\}} \mathcal{N}(\boldsymbol{y}_k^q | \bar{\boldsymbol{y}}_k^{\boldsymbol{r}_k^q}, \boldsymbol{S}^{\boldsymbol{r}_k^q}) \right]$$

where $\bar{\boldsymbol{y}}_k^{\boldsymbol{r}_k^q}$ is the predicted mean (using the measurement model) and $\boldsymbol{S}^{\boldsymbol{r}_k^q}$ is the innovation covariance dependent of the previous existence as determined by the association event $\boldsymbol{r}_k^q$, and 1_k^n is a target indicator function. After having this, the *update* is done as a bank of Kalman filters where there is an adapted innovation for each measurement.

The implementation for the Gaussian JPDA used in this work is a modification for tracking on videos using the toolbox by Särkkä et al. [43].

3.5 Markov Chain Monte Carlo Framework

Markov Chain Monte Carlo (MCMC) data association is an extension of the JPDA approach to allow a varying number of targets. As we pointed out, the JPDA tackles uncertain data association conditions by allowing a target to be updated by a weighted sum of all measurements determined to be within a distance threshold. The weights represent the probability that the measurement originates from that target. As such, a measurement can contribute to more than one track, and it contributes to this target with a certain weight. MCMC data association expands on this by considering the space of all possible associations, where each association event has three possible moves: deletion, addition (birth) or survival [44]. The weights are calculated similarly to the JPDA but Monte Carlo methods (such as Metropolis-Hastings [28]) are used to integrate over the set and evaluate the probability of each of the moves.

The solution set $\mathbf{\Lambda}_{t_k}$ contains association event λ_{t_k} histories over multiple steps or a single step as stated on the original literature [44], as well as considering all possible numbers of targets at each time step. The association event λ_{t_k} contains a set of $\bar{Q}(t_k)$ in total, with F_k false elements and a set of ν_k measurement from new targets and the remaining measurements $\bar{Q}(t_k) - F_k - \nu_k$ come from existing targets. In the presentation for the JPDA we introduced the weights c_{t_k} that are calculated according to Eq. (3.4.7). MCMC data association extends this to include the birth of new targets as seen on the composition of the association events. In order to do this, the posterior of the the association events is represented as

$$p(\lambda_{t_k}|\boldsymbol{y}_{1:k}) = \prod_{i=0}^{F_k} p_C \prod_{i=0}^{\nu_k} p_B \prod_{i=0}^{\bar{Q}(t_k)-F_k-\nu_k} p(\boldsymbol{y}_k|\boldsymbol{\Lambda}_{t_k}, \boldsymbol{y}_{1:k-1}) \tag{3.5.1}$$

where p_C is the assumed clutter density, p_B is the birth density, and $p(\boldsymbol{y}_k|\boldsymbol{\Lambda}_{t_k}, \boldsymbol{y}_{1:k-1})$ is the likelihood of the measurement given the past measurements and existing targets. The final representation depends of the assumptions and in general has the same form than Eq. (3.4.7) adding in this case the birth density.

$$\begin{aligned} p(\lambda_{t_k}|\boldsymbol{y}_{1:k}) \quad \propto \quad & V^{-F_k} \frac{F_k!}{\bar{Q}(t_k)!} P(F)(P_k^{det})^{1_k}(1-P_k^{det})^{1-1_k} \\ & (P_k^{del})^{1_k^{del}}(1-P_k^{del})^{1-1_k^{del}} \int p(\boldsymbol{y}_k|\boldsymbol{x}(t_k))p(\boldsymbol{x}(t_k)|\boldsymbol{y}_{1:k-1})d\boldsymbol{x}(t_k). \end{aligned} \tag{3.5.2}$$

The MCMC procedure then consists on using Eq. (3.5.3) to integrate and search the space $\boldsymbol{\Lambda}_{t_k}$. This is achieved through the Monte Carlo integration technique known as Metropolis-Hastings algorithm using Eq. (3.5.3) as the proposal sampling distribution to evaluate the possible moves for each association event. This procedure features efficient mechanisms to search over this large solution space in addition to birth and death moves to add or remove targets (Section IV.A in [44]). For a general description of the Metropolis-Hastings algorithm and Monte Carlo integration techniques see [28].

After introducing the MCMC data association it is important to point out that there are other Monte Carlo methods that have been used as a solution for the problem of multiple target tracking but with a batch processing approach. Important examples are the reversible jump MCMC and the particle MCMC, both based on the standard MCMC described above. Their description is outside the scope of this work since we focus on the

sequential approaches, but details for multiple target tracking applications and parameter estimation can be found in [45].

3.5.1 Gaussian Approximation

The effects of the Gaussian assumption for measurement and motion models is identical to the effects within the JPDA technique. The approximation of the joint target state $p(\boldsymbol{X}(t_k)|\boldsymbol{Y}_k)$ is done by a factorial form where each factor is the marginal distribution $p(\boldsymbol{x}^n(t_k)|\boldsymbol{y}_{1:k})$ corresponding to a single target, then obtaining the representation

$$p(\boldsymbol{x}^n(t_k)|\boldsymbol{y}_{1:k}) \approx \mathcal{N}(\boldsymbol{x}^n(t_k)|\boldsymbol{\mu}^n(t_k), \boldsymbol{\Sigma}^n),$$

where $\boldsymbol{\mu}^n(t_k)$ and $\boldsymbol{\Sigma}^n$ are mean and covariance of the target n. The Gaussian assumption of the transition and likelihood models reduces the *prediction* step to a regular Kalman filter acting on each of the targets separately. Finally the proposal for the Monte Carlo search methods is

$$\begin{aligned} p(\lambda_{t_k}|\boldsymbol{y}_{1:k}) \quad \propto \quad & V^{-F_k}\frac{F_k!}{\bar{Q}(t_k)!}P(F)(P_k^{det})^{1_k}(1-P_k^{det})^{1-1_k} \\ & (P_k^{del})^{1_k^{del}}(1-P_k^{del})^{1-1_k^{del}}\mathcal{N}(\boldsymbol{y}_k|\bar{\boldsymbol{y}}_k^{\boldsymbol{r}_k}, \boldsymbol{S}^{\boldsymbol{r}_k}). \end{aligned} \tag{3.5.3}$$

The implementation for the Gaussian MCMC data association framework used in this work is a modification for tracking on videos using the toolbox by Särkkä et al. [43].

3.6 Probability Hypothesis Density Filter

This is one of a set of techniques based on the random finite sets framework. The general multiple target tracking presented in Section 3.1 shows the general Bayesian representation that is also used by the random finite sets framework, but the sets involved are categorized as random finite sets. A random finite set is usually defined as sets composed by random variables that at the same time have a random cardinality. It can be completely described by a discrete distribution that characterizes the cardinality and probability density functions that describe the points inside the set, conditional or not of the cardinality [2; 46].

The different techniques that arise from the random finite set framework come from the way in which Eq. (3.1.6) is approximated to be tractable according to a recent field of statistics known as finite set statistics (FISST) [2]. According to FISST the first moment of a finite random set is known as the probability hypothesis density (PHD) and the filtering technique resulting from this approximation has the following structure [47]:

- **PHD time update:** Given the process model, the predicted PHD,

$$D_{k|k-1}(\mathrm{x}(t_k)|\mathrm{Y}_{k-1}) = \overbrace{\gamma_k(\mathrm{x}(t_k))}^{\text{born targets}} + \int \overbrace{p_S(\mathrm{x}(t_{k-1})).f_{k|k-1}(\mathrm{x}(t_k)|\mathrm{x}t(_{k-1}))}^{\text{existing targets}} \\ .D_{k-1|k-1}(\mathrm{x}(t_{k-1})|\mathrm{Y}_{k-1})d\mathrm{x}(t_{k-1}) \tag{3.6.1}$$

  where,

  - $D_{k|k-1}$ PHD of all the targets after prediction.
  - $\gamma_k(\mathrm{x}(t_k))$: PHD of the new targets.
  - $p_S(\mathrm{x}(t_{k-1}))$: Probability of a target being detected.

- **PHD data update:** Given a new set of measurements Y_k, the updated PHD,

$$\begin{aligned} D_{k|k}(\mathrm{x}(t_k)|\mathrm{Y}_k) &= (1-p_D)D_{k|k-1}(\mathrm{x}(t_k)|\mathrm{Y}_{k-1}) \\ &+ \sum_{\mathrm{y}_k \in \mathrm{Y}_k} \frac{p_D D_k(\mathrm{y}_k)}{\lambda_c c_k(\mathrm{y}_k) + p_D D_k(\mathrm{y}_k)} D_k(\mathrm{x}(t_k)|\mathrm{y}_k) \end{aligned} \tag{3.6.2}$$

  where,

$$D_k(\mathrm{y}_k) = \int p(\mathrm{y}_k|\mathrm{x}(t_k)) D_{k|k-1}(\mathrm{x}(t_k)|\mathrm{Y}_{k-1}) d\mathrm{x}(t_k) \tag{3.6.3}$$

$$D_k(\mathrm{x}(t_k)|\mathrm{y}_k) = \frac{p(\mathrm{y}_k|\mathrm{x}(t_k)) D_{k|k-1}(\mathrm{x}(t_k)|\mathrm{Y}_{k-1})}{D_k(\mathrm{y}_k)} \tag{3.6.4}$$

  and,

  - $p(\mathrm{y}_k|\mathrm{x}_k)$: is the sensor likelihood function $L_{\mathrm{y}}(\mathrm{x}(t_k))$
  - λ_c: average number of false alarms per scan, which is assumed to be Poisson distributed
  - $c_k(\mathrm{y}_k)$: distribution of each of the false alarms

This technique makes use of the general assumptions in **DA**.**1a** and **DA**.**2a**, which means that all the targets and measurements are processed at the same time without performing a data association step. In order to extend this technique for labeling of the targets it is necessary to have extra algorithms in order to manage target labeling. The mathematical details related to the presented structure can be seen in the work by Vo et al. [47; 2] and its presentation is outside the scope of this work.

3.6.1 Gaussian Approximation

Under Gaussian assumptions the PHD filter is known as the Gaussian mixture models PHD (GMMPHD) [47] and has been widely used in the literature [2; 5; 46; 48]. It mainly consists on assuming that motion model, likelihood function, and birth density are assumed Gaussian, and presents the following structure:

Initialize

At time $t_k = 0$, the PHD $D_{k|k}$ is initialized with a weighted sum of J_k Gaussians

$$D_{k|k}(\mathrm{x}|\mathrm{Y}_k) = \sum_{j=1}^{J_k} w_k^{[j]} \mathcal{N}(\mathrm{x}; \mu_k^{[j]}, \Sigma_k^{[j]})$$

These are distributed across the state space where each Gaussian term $\mathcal{N}(\mathrm{x}; \mu_k^{[j]}, \Sigma_k^{[j]})$ has a corresponding weight $w_k^{[j]}$, mean $\mu_k^{[j]}$, and variance $\Sigma_k^{[j]}$. At $k \geq 1$,

PHD Time Update The predicted PHD up to time k is a Gaussian mixture,

$$D_{k|k-1}(\mathrm{x}) = D_{S,k|k-1}(\mathrm{x}) + \gamma_k(\mathrm{x})$$

where, $D_{S,k|k-1}(\mathrm{x})$ is predicted intensity of the existing (survived) objects in the FOV of the sensor, given by,

$$D_{S,k|k-1}(\mathrm{x}) = p_S \sum_{j=1}^{J_{k-1}} w_{k-1}^{[j]} \mathcal{N}(\mathrm{x}; \mu_{S,k|k-1}^{[j]}, \Sigma_{S,k|k-1}^{[j]})$$

with,

$$w_{k|k-1}^{[j]} = p_S . w_{k-1}^{[j]}; \quad \mu_{S,k|k-1}^{[j]} = F_{k-1} \mu_{k-1}^{[j]};$$

$$\Sigma_{S,k|k-1}^{[j]} = Q_{k-1} + F_{k-1} \Sigma_{k-1}^{[j]} F_{k-1}^T$$

and, $\gamma_k(\mathrm{x})$ is the PHD representing the new incoming targets, given by,

$$\gamma_k(\mathrm{x}) = \sum_{j=1}^{J_{\gamma,k}} w_{\gamma,k}^{[j]} \mathcal{N}(\mathrm{x}; \mu_{\gamma,k}^{[j]}, \Sigma_{\gamma,k}^{[j]})$$

with,

$$w_{k|k-1}^{[j]} = w_{\gamma,k}^{[j]}; \quad \mu_{k|k-1}^{[j]} = \mu_{\gamma,k}^{[j]}; \quad \Sigma_{k|k-1}^{[j]} = \Sigma_{\gamma,k}^{[j]}$$

PHD Data Update The PHD measurement update is a Gaussian mixture given by,

$$D_{k|k}(\mathrm{x}) = (1 - p_D) D_{k|k-1}(\mathrm{x}) + \sum_{z \in \mathrm{Z}_k} D_{L,k}(z|\mathrm{x})$$

where,

$$D_{L,k}(z|\mathrm{x}) = \sum_{j=1}^{J_{k-1}+J_{\gamma,k}} w_{k|k}^{[j]} \mathcal{N}(\mathrm{x}; \mu_{k|k}^{[j]}, \Sigma_{k|k}^{[j]})$$

with the standard Kalman filter update for each component of the mixture model,

$$w_{k|k}^{[j]} = \frac{p_D w_{k|k-1}^{[j]} f_k^{[j]}(z|\mathrm{x})}{\lambda_c c_k(z) + \sum_{l=1}^{J_{k-1}+J_{\gamma,k}} w_{k|k-1}^{(l)} f_k^{(l)}(z|\mathrm{x})}$$

$$f_k^{[j]}(z|\mathrm{x}) = \mathcal{N}(z; H_k \mu_{k|k-1}^{[j]}, S_k^{(i)});$$

$$\mu_{k|k}^{[j]} = \mu_{k|k-1}^{[j]} + K_k^{[j]}[z - H_k \mu_{k|k-1}^{[j]}]; \Sigma_{k|k}^{[j]} = [I - K_k^{(j)} H_k^{[j]}] \Sigma_{k|k-1}^{[j]};$$

$$K_k^{[j]} = \Sigma_{k|k-1}^{[j]} [H_k^{[j]}]^T [S_k^{(j)}]^{-1}; S_k^{[j]} = R_k + H_k^{[j]} \Sigma_{k|k-1}^{[j]} [H_k^{[j]}]^T$$

Thus, there are $J_k = (1 + |\mathrm{Y}_k|)(J_{k-1} + J_{\gamma,k})$ Gaussian components in the updated PHD with $(1 + |\mathrm{Y}_k|)$ components for each prediction term at time k and the Gaussian mixture is of the form,

$$D_{k|k}(\mathrm{x}) = \sum_{j=1}^{J_k} w_{k|k}^{[j]} \mathcal{N}(\mathrm{x}; \mu_k^{[j]}, \Sigma_k^{[j]})$$

The implementation in this work comes from Bryan Clarke[1] that is based on the seminal work [47].

[1]Found in http://www.mathworks.com/matlabcentral/fileexchange/42769-gaussian-mixture-probability-hypothesis-density-filter--gm-phd-

CHAPTER 4

Finite Point Processes and Multiple Target Tracking transform

Given the set of multiple target tracking techniques introduced in the previous chapter, it is important to introduce the framework that is ultimately used to encompass all the techniques, and can be used as the tool to extract the information content of the different assumptions and elements of the techniques.

This chapter presents all the concepts for Finite processes and the representation of target tracking concepts and techniques as probability generating functions and functionals. Finally it presents the conceptualization of probability generating functionals as a MTT transform.

4.1 Introduction to Finite Point Process

Point processes are usually introduced in the framework of the theory of random measures.

Let $\boldsymbol{\chi}$ be a 'nice' topological space (more precisely, a complete, separable, metric space) A typical choice for $\boldsymbol{\chi}$ is $\mathbb{R}^d$, $d > 0$. The space of sets of points or event space in $\boldsymbol{\chi}$ is defined by

$$\boldsymbol{\varepsilon}_{\boldsymbol{\chi}} = \emptyset \cup \bigcup_{n \geq 1} \boldsymbol{\chi}(n), \tag{4.1.1}$$

where $\boldsymbol{\chi}(n)$ is the space of sets of size $n \in \mathbb{N}$, that is

$\boldsymbol{\chi}(n) = \big\{\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} | \boldsymbol{x}_i \in \boldsymbol{\chi}, i = 1, \ldots, n\big\}$. All its elements are assumed to be locally finite and each bounded subset of $\boldsymbol{\chi}$ can contain only a finite number of points [49].

Although a point process is regarded as a random (multi)set $\{\boldsymbol{x}_i\}_i \subset \boldsymbol{\chi}$, it is technically convenient to formally define it as a random measure $\xi = \sum_i \delta_{\boldsymbol{x}_i}$.

$$\Phi : (\Omega, \mathcal{F}, \mathbb{P}) \to (\varepsilon_\chi, B(\varepsilon_\chi)) \tag{4.1.2}$$

where $(\Omega, \mathcal{F}, \mathbb{P})$ is an arbitrary probability space and $B(\varepsilon_\chi)$ denotes the Borel σ-algebra of ε_χ. Hence, if Φ denotes the point process $\{\phi_i\}$, we write $\Phi(A)$ for the number of points ϕ_i that belong to a subset $A \subseteq \boldsymbol{\chi}$; similarly, for suitable functions f on $\boldsymbol{\chi}$, $\int f \, d\Phi = \sum_i f(\phi_i)$.

Thus, let $\mathfrak{N} = \mathfrak{N}(\boldsymbol{\chi})$ be the class of all Borel measures μ on $\boldsymbol{\chi}$ such that $\mu(A)$ is a (finite) integer $0, 1, \ldots$ for every relatively compact Borel set A; this coincides with the class of all finite or countably infinite sums of the type $\sum_i \delta_{x_i}$, where $x_i \in \boldsymbol{\chi}$ and each compact subset of $\boldsymbol{\chi}$ contains only a finite number of x_i, and we identify such a sum with the (multi)set $\{x_i\}$.

A point process on $\boldsymbol{\chi}$ is a random element of $\mathfrak{N}$. If Φ is a point process on $\boldsymbol{\chi}$, there exists a unique Borel measure ν on $\boldsymbol{\chi}$ such that $\mathbb{E}\Phi(A) = \nu(A)$ for every Borel set A, and more generally $\mathbb{E} \int h \, d\Phi = \int h \, d\nu$ for every positive measurable function h. This measure ν is called the *intensity* of Φ and it completely statistically describes it.

Another descriptor of the point process comes from the finite-dimensional distributions ('fidi') of a random measure ξ that are the joint distributions, for all finite families of bounded Borel sets $A_1, \ldots, A_k$ of the random variables $\xi(A_1), \ldots, \xi(A_k)$, that is the image of the probability measure $\mathbb{P}$, represented as P_Φ [50].

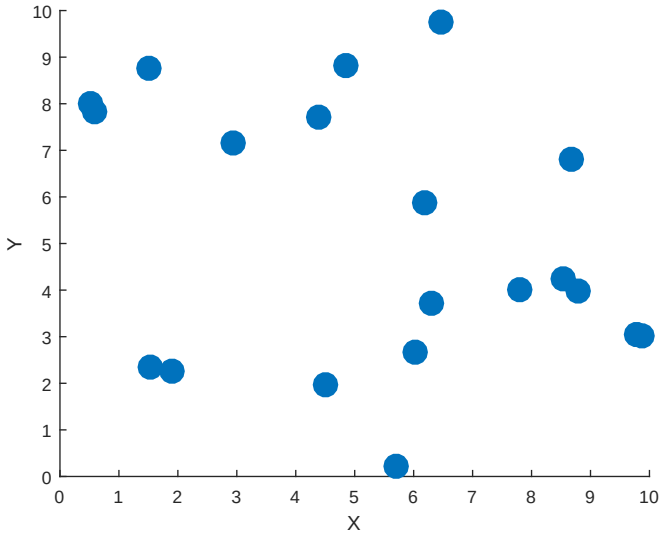


Figure 4.1: Example of a point process in 2D

4.1.1 Probability Generating Functional

The information about a point process can be encoded in an algebraic expression (functional) similar to the manner done for continuous and discrete dynamical systems (laplace and z-transforms), sequence of number and vectors (generating functions for combinatorics analysis) and probability density functions (Probability generating functions). Before introducing the mathematical definition of probability generating functionals it is important to introduce the most commonly known concept of probability generating functions that are also useful in the context of the research at hand.

Given a random variable X on the space $(X, B_{\boldsymbol{\chi}})$ the probability generating function (p.g.f.) of X is the function defined, for each $z \in \mathbb{R}$, as $\mathbb{E}[z^X]$.

$$G(z) = E[z^X] = \sum_{x=0}^{\infty} p(x) z^x \tag{4.1.3}$$

This of course can be extended to multiple dimensions where $(z_1, \ldots, z_n) \in \mathbb{R}^n$. In order to introduce the general mathematical formulation of the probability generating functional (p.g.fl.) it is important to introduce $V(\boldsymbol{\chi})$ the set of $B_{\boldsymbol{\chi}}$ -measurable (test) functions $h : \boldsymbol{\chi} \to R$ such that $1 - h(x)$ vanishes out of some bounded set and $0 \leq h(x) \leq 1$ for each $\boldsymbol{x} \in \boldsymbol{\chi}$, with this the p.g.fl. of a general point process Φ on the space $\boldsymbol{\chi}$ is defined for each $h \in V(\boldsymbol{\chi})$ [49], as

$$\Psi[h] \equiv \Psi_{\Phi}[h] = \mathbb{E}\left[exp\left(\int_{\chi} \log[h(\boldsymbol{x})]\Phi(d\boldsymbol{x})\right)\right] \tag{4.1.4}$$

Since the process Φ is defined to be finite on the set where $1 - h(\boldsymbol{x}) \neq 1$ then it can be written as

$$\Psi[h] \equiv \Psi_{\Phi}[h] = \mathbb{E}\left[\prod_{i=1} h(\boldsymbol{x}_i)\right] \tag{4.1.5}$$

where $\boldsymbol{x}_i$ are the points such that $\Phi = \sum_i \delta_{\boldsymbol{x}_i}$, possibly having repetitions in the (multi)set $\{x_i\}$. In order to realize this expected value is important to remember the fidi of the point process which allows us to rewrite the p.g.fl [51] as

$$\Psi[h] \equiv \sum_{n\geq 0} \int_{\chi(n)} \prod_{i=1}^{n} h(\boldsymbol{x}_i) P_{\Phi}(d\{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}) \tag{4.1.6}$$

$$= \sum_{n\geq 0} \frac{1}{n!} \int_{\chi^n} \prod_{i=1}^{n} h(\boldsymbol{x}_i) p_n(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n) d\boldsymbol{x}_1 \cdots d\boldsymbol{x}_n \tag{4.1.7}$$

where the final representation is obtained thanks to the combinatorics interpretation of a

Janossy measure [51] applied to the fidi. This representation can then be extended to joint point processes, where a new process Υ is introduced, with similar characteristics to Φ but on space $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{d_y}$ (in this application it can considered as the measurement space and it has a mapping to the state space). Thus, Eq. (4.1.7) can be extended to the joint space and defined on $\varepsilon_\chi \times \varepsilon_\Upsilon$ as the products of the random measures

$$\begin{aligned} \Psi_{\Phi\Upsilon}[g,h] \equiv & \sum_{m\geq 0}\sum_{n\geq 0}\frac{1}{m!n!}\int_{\Upsilon^m}\int_{\chi^n}\prod_{i=1}^{m} g(\boldsymbol{y}_i)\prod_{i=1}^{n} h(\boldsymbol{x}_i) \\ & p_{\Phi\Upsilon}(\boldsymbol{y}_1,\cdots,\boldsymbol{y}_m,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n)d\boldsymbol{y}_1\cdots d\boldsymbol{y}_m d\boldsymbol{x}_1\cdots d\boldsymbol{x}_n \end{aligned} \tag{4.1.8}$$

Marginalizing this p.g.fl with respect to one process results in the p.g.fl of the other process.

$$\Psi_{\Phi\Upsilon}[1,h] = \Psi_\Phi[h] \qquad \text{and} \qquad \Psi_{\Phi\Upsilon}[g,1] = \Psi_\Upsilon[g] \tag{4.1.9}$$

4.2 Point Processes and Target Tracking

At this point it is important to talk about the relationship of target tracking (previous chapters) and the concepts introduced so far in this chapter to introduce the fundamental conceptual connections that make up the most important contributions of this work. First, it is important to note the big similarity between the general spaces described in section 3.1 and earlier in section 4.1 (a third definition with similar characteristics comes from the random finite sets framework [2], briefly mentioned in chapter 3 to introduce the PHD filter). This similarity has prompted a body of research [2] [52] [3] [53] on the subject of finding a framework that encompasses all the MTT techniques and gives space to the

birth of newer and theoretically grounded techniques.

The theory of Point Processes has been used for statistical analysis of data in many applications, for instance [54] [50] [2] [49], but it started being linked with the classical MTT techniques through the early developments of random finite sets statistics [55] [2] and finally through the work by Streit et al. [53][56] [52]

4.2.1 Application of the P.G.FLs for Tracking

The expressions and theory just presented has only been around for a short period of time, and so far it has been only used to do tracking with the JPDA [56]. It has also been used a framework to show the closed form solution for the PHD filter and the introduction of the intensity filter [52].

4.2.2 Information Encoding and Generating Functions

A generating function is an algebraic tool for encoding combinatorial data. With this in mind we can claim that given the p.g.f and p.g.fls encode all the combinational information of the finite point process they represents. If it is assumed for example that a tracking technique is a finite point process, then its p.g.fl representation encodes all the information pertinent to target and measurement existence and the set of assumption encoded with the technique.

4.3 Probability Generating Functionals and Generating Functions for Target Tracking

This section is an application of the p.g.fs and p.g.fls to target tracking in general as described in the work by Streit et al. [53] and it is included for completeness of the presentation but not as an original work of the author.

4.3.1 Target Detection

In the general literature missed detections are represented as a binary decision that translate into the p.g.f of a binomial distribution. The difference comes from the interpretation of the coefficient on the p.g.f that will depend on the probability of detection $P_k^D(\boldsymbol{x})$ of an object known to be present at state $\boldsymbol{x} \in \boldsymbol{\chi}$ at an instant of time t_k. Then the coefficient $a(\boldsymbol{x})$ is the probability of miss detection $1 - P_k^D(\boldsymbol{x})$ and $b(\boldsymbol{x})$ is the previously mentioned probability of detection, obtaining the p.g.f

$$G_{M|\boldsymbol{x}}^{BMD}(z) \equiv a(\boldsymbol{x}) + b(\boldsymbol{x})z \tag{4.3.1}$$

This can be modified to the detection of extended targets but it is not part of the applications described in this work (details in [53]).

4.3.2 Clutter Models

As was mentioned in the chapter 3 the common assumption for clutter in the tracking environment is a poisson distribution of objects in the scenario. The effect of clutter is seen

in the space of the measurement $\boldsymbol{\mathcal{Y}}$ and are represented as a Poisson point process [3]

$$\Psi_C^{PPP}[g] = exp\left(-\Lambda + \Lambda \int_{\boldsymbol{Y}} g(\boldsymbol{y}) p_\Lambda(\boldsymbol{y}) d\boldsymbol{y}\right) \tag{4.3.2}$$

where Λ is mean number of clutter points in $\boldsymbol{\mathcal{Y}}$ and $p_\Lambda(\boldsymbol{y})$ is the normalized intensity function. For tracking application these functions are usually assumed constant in the window of observation.

4.3.3 Bayes-Markov Filters with Miss Detection

The Bayes-Markov p.g.fl encodes all the information presented in the single target tracking general bayesian. Presenting the usual bayesian prediction-update form represented on the target/measurement existence plane.

$$\Psi_{BM}[g,h] \equiv \int_{\boldsymbol{S}} \int_{\boldsymbol{Y}} h(\boldsymbol{x}) g(\boldsymbol{y}) \mu(\boldsymbol{x}) p(\boldsymbol{y}|\boldsymbol{x}) d\boldsymbol{y} d\boldsymbol{x} \tag{4.3.3}$$

If $\Psi_{BM}[g,h]$ is used as the argument in Eq. (4.3.1) the combinatorics representation of taking in account miss detection probabilities, obtaining

$$\begin{aligned} \Psi_{BMD}[g,h] &\equiv \int_{\boldsymbol{S}} h(\boldsymbol{x}) \mu(\boldsymbol{x}) G_{M|\boldsymbol{x}}^{BMD}\left(\int_{\boldsymbol{Y}} g(\boldsymbol{y}) p(\boldsymbol{y}|\boldsymbol{x}) d\boldsymbol{y}\right) d\boldsymbol{x} & (4.3.4)\\ &= \int_{\boldsymbol{S}} h(\boldsymbol{x}) \mu(\boldsymbol{x}) \left(a(\boldsymbol{x}) + b(\boldsymbol{x}) \int_{\boldsymbol{Y}} g(\boldsymbol{y}) p(\boldsymbol{y}|\boldsymbol{x}) d\boldsymbol{y}\right) d\boldsymbol{x}. & (4.3.5) \end{aligned}$$

The different coefficients represent the probability of a target state $\mu(\boldsymbol{x})$, likelihood of

the measurements $p(\boldsymbol{y}|\boldsymbol{x})$, probability of detection $b(\boldsymbol{x})$ and probability of miss-detection $a(\boldsymbol{x})$. $\Psi_{BMD}^{Data}[g,h]$ has the same representation but it uses a given birth density instead of $\mu(\boldsymbol{x})$. When evaluated under Gaussian assumption, it presents the structure of the a standard Kalman filter with miss detection, sometimes called the Probabilistic Data Association filter when clutter is added in the assumptions.

4.3.4 JPDA

This is an extension of the Bayes-Markov filter with miss detection for a known number of targets n, and taking in account the probability of having false alarms from clutter in the environment (section 4.3.2)

$$\Psi_{JPDA}[g,h_1,\ldots,h_n] = \Psi_C^{PPP}[g]\prod_{i=1}^{n}\Psi_{BMD(i)}[g,h_i]. \tag{4.3.6}$$

In this case there are as many target existence functionals h as targets. This means that each of the has its own state space and can only produce one measurement and have the need of performing data association (implying the assumption described in chapter 2 for this technique).

4.3.5 MHT and MCMC framework

MCMC is really similar in nature to the JIPDA introduced in [53] but it has not been explicitly introduced before though it was briefly mentioned in the discussion of [56]. It presents an extension of the JPDA for an unknown number of targets, where existing targets and birthed targets are taking in account. The way in which the JPDA is extended is by

using the p.g.f for the number of targets

$$G_N[z] = 1 - \chi + \chi z \tag{4.3.7}$$

where χ represents the updated probability of target existence. Using Ψ_{BMD} as an argument for Eq. (4.3.7) and superposing existing targets, birthed targets and clutter, the p.g.fl for the MCMC looks like

$$\Psi_{MCMC}[g, h_1, \ldots, h_{n+m}] = \Psi_C^{PPP}[g] \prod_{i=1}^{n} \left[1 - \chi_i + \chi_i \Psi_{BMD(i)}[g, h_i]\right] \times \prod_{j=1}^{m} \left[1 - \gamma_j + \gamma_j \Psi_{BMD(j)}^{Data}[g, h_i]\right]. \tag{4.3.8}$$

where χ_i is the probability of existence, and γ_j is the probability of birth from acquired data respectively. For the MHT we have the same expression as introduced by Streit et al. [53]

$$\Psi_{MHT}[g, h_1, \ldots, h_{n+m}] = \Psi_C^{PPP}[g] \prod_{i=1}^{n} \left[1 - \chi_i + \chi_i \Psi_{BMD(i)}[g, h_i]\right] \times \prod_{j=1}^{m} \left[1 - \gamma_j + \gamma_j \Psi_{BMD(j)}^{Data}[g, h_{n+j}]\right] \tag{4.3.9}$$

again χ_i is the probability of existence, and γ_j is the probability of birth from acquired data respectively. The difference comes from the way in which the values for χ and γ are calculated. In the case of the MHT both birth and existence are evaluated with the hypothesis evaluation equations introduced in chapter 3, and for the MCMC are evaluated depending the Monte Carlo Jumps for the equations presented in chapter 3.

4.3.6 PHD

This was the first technique introduced through the use of the point process framework [52] besides its original random finite sets origins introduced in chapter 3, where it was shown that the PHD filter is the intensity of a Poisson point process. The p.g.fl then represent the superposition of the clutter p.g.fl and the p.g.fl of a Poisson point process for the number of targets with the Bayes-Markov filter as its argument

$$\Psi_{PHD}[g,h] = \Psi_C^{PPP}[g]\Psi_N^{PPP}\big[\Psi_{BMD}[g,h]\big] \tag{4.3.10}$$

One of the main differences with the other techniques described so far with point process framework, comes from the fact that there is only one target existence functional h since it is assumed that all targets move in one unique state space, which in practice allows to have several measurements per target.

4.4 Probability Generating Functional as the Multiple Target Transform

The idea of a Multiple Target transform comes from the fact that a generating function is an algebraic tool for encoding combinatorial data [57]. With this in mind, we can claim that the given p.g.fs and p.g.fls encode all the combinational information of the finite point process they represent. If it is assumed that a tracking technique is a joint finite point process [53], then its p.g.fl representation encodes all the information pertinent to target and measurement existence and the set of assumptions encoded within the technique. It is also a fact that the test functions in which a p.g.fl is evaluated are complex numbers (like any classical transform), that in this case represent a space of "existence" for each target (on a

Table 4.1: MTT marginalized transforms

Technique	**Marginalization**
JPDA	$\Psi_C^{PPP}[g]\Psi_{BMD}[g]^n$
MHT	$\Psi_C^{PPP}[g]\left(1-\chi+\chi\Psi_{BMD}[g]\right)^n\left(1-\gamma+\gamma_i\Psi_{BMD}^{Data}[g]\right)^m$
MCMC	$\Psi_C^{PPP}[g]\left(1-\chi+\chi\Psi_{BMD}[g]\right)^n\left(1-\gamma+\gamma_i\Psi_{BMD}^{Data}[g]\right)^m$
PHD	$\Psi_C^{PPP}[g]\Psi_N^{PPP}\left[\Psi_{BMD}[g]\right]$

z-transform for example it represents a discrete time delay). This means that if a target exists there is a complex number representing its existence and the same occurs for measurements.

As described in Section 4.2.1, in order to use p.g.fl's for the estimation of the state of the targets it is necessary to evaluate all these complex variables (that vary on dimension too) as can be seen in [56], which is the equivalent of finding the inverse transform for classical techniques. In our case, the objective is to predict the performance of the technique, so therefore we want to avoid performing all the estimation process. Here is where our use of the concepts from p.g.fl's gives us some insight on how we can achieve a new simplified representation for this purpose.

4.4.1 Marginalized Transform

Given that the p.g.fl for the techniques represent the point precesses of joint spaces, target(s) and measurements, the introduction of this simplified representation comes from the use of Eq. (4.1.9), assuming targets existence where there are measurements present. The obtained expressions can be seen in Table 4.1.

Since we marginalized over target existence this represents all the information encoded on each measurement including much more than only the likelihood, giving us a measure of the

amount of information that can be perceived for a set of measurements on a given instant of time.

CHAPTER 5

Tracker Quality Assessment and Prediction

This chapter gives a summary of the concepts and work done on image quality and video quality assessment in the literature. It also introduces the frameworks that uses the formerly introduced MTT transform to obtain a quantity that will be used as a tracker quality assessment.

5.1 Background on Image/Video Quality Assessment

In general, the field of image and video processing focuses on signals that are meant for human consumption, including images and videos presented over the Internet. Before an image or video is transmitted to a human observer, it may undergo many stages of processing that can introduce distortions that reduce the quality of the final display. For example, camera devices can potentially introduce distortions into an image or video due to the recording devices optics, sensor noise, color calibration, exposure control, and motion. Following its acquisition, the image or video may be processed further by a compression algorithm that reduces the bandwidth requirements for storage or transmission. These compression algorithms are predominantly designed to maximize savings in bandwidth by allowing certain distortions to affect the signal. Similarly, bit errors, which can occur while an image is being transmitted over a channel or (infrequently) when it is stored, also tend to present distortions. Additionally, the display device that renders the final output may cause distortions, such as low reproduction resolution or bad calibration. At each stage of

processing, the level of distortion that occurs mainly depends on economics and/or physical limitations of the camera devices.

Research in objective image quality assessment seeks to design quantitative measures with the capability to automatically predict perceived image quality. Once developed, an objective image quality metric can be applied to an extensive range of practical uses. These applications include image acquisition, compression, communication, displaying, printing, restoration, enhancement, analysis, and watermarking. To accomplish this, an objective image quality metric can incorporate the following three processes: (i) active monitoring and adjustment of image quality, (ii) optimization of algorithms and parameter settings of image processing systems, and (iii) benchmarking of image processing systems and algorithms [58].

To summarize, objective quality measurement of images offers an algorithmic determination of image and video quality instead of relying on subjective human observations. The aim of research in this area of quality assessment (QA) is to develop algorithms whose quality prediction aligns with subjective assessments from human observers [59].

5.2 Objective Quality Methods

There are two classifications of objective quality methods: psychophysical and engineering approaches. Psychophysical metrics attempt to model the human visual system (HVS) by incorporating aspects like contrast and orientation sensitivity, frequency selectivity, spatial and temporal pattern, masking, and color perception. Although these metrics can correct a variety of video degradations, the computations are typically demanding. The engineering approach utilizes simplified metrics based on the extraction

and analysis of certain features and artifacts in a video. This approach does not necessarily neglect the attributes of the HVS, for it still considers psychophysical factors. However, the engineering metrics are based on analysis of video content and distortion instead of human visual modeling.

In order to create an objective quality method that can generate a mean opinion score, a set of features or quality-related parameters of the image or video are gathered. These objective quality methods are further classified by the degree of reference information available from the original image or video. The categories of objective quality methods include full reference (FR), reduced reference (RR), and no reference (NR), with the criteria for each indicated as follows [60]:

- FR methods: The entirety of the original image or video is available as a reference. Thus, FR methods compare the distorted image or video with the original.
- RR methods: Requires the provision of representative features about texture or other characteristics of the original image or video, rather than access to the complete original. Consequently, the input for RR methods is the comparison of the reduced information from the original image or video with the analogous information from the corresponding distorted image or video.
- NR methods: Access to the original image or video is not required. Instead, the comparison is based upon certain artifacts relating to the pixel domain of an image or video, the information embedded in the bitstream of the image or video format, or a hybrid of both.

5.3 Analysis of the Decoded Video

Due to the fact that video quality metrics analyze the decoded video in distinct ways, they can be divided into three classes: data metrics, picture metrics, and packet- and bitstream-based metrics.

5.3.1 Data Metrics

In this class, the fidelity of the video signal is assessed without modeling any element of the HVS. Commonly used data metrics in video quality evaluation are mean square error (MSE) and its logarithmic representation peak signal-to-noise ratio (PSNR), as they are simple to understand and implement. However, these metrics generally do not yield an objective quality measure that corresponds well with the perceptions of a human observer for a variety of coding and transmission parameters. This disparity can be explained by the fact that data metrics compare the reference and test data while neglecting the concept of what the data actually represents. By not considering HVS characteristics, these metrics demonstrate the differences between them and the HVS regarding sensitivities to distortion types and properties. Furthermore, the MSE/PSNR does not take into account the specific location that distortion appears in a frame, which can be important in evaluating video quality.

Bit error rate (BER) and packet loss rate (PLR) are data metrics used for videos transmitted over the Internet. When using BER and PLR, the same problem that appears in the MSE/PSNR approach occurs because they do not account for the content of the packet and its influence on visual quality. BER and PLR assign the same visual importance to all packets, which does not adequately provide for video delivery. Consequently, they

effectively measure the percentage of incorrect bits or lost packets, but they cannot evaluate perceived video quality.

5.3.2 Picture Metrics

To address the problems that arise while using data metrics, various objective video quality metrics attempt to predict the perceived video quality. In order to do this, they consider information about the video content and distortion types, typically by analyzing the visual information within the video data. These metrics are called picture metrics [61]. Depending on the approach employed in the metric design, picture metrics are divided into two classes: (i) a vision modeling approach and (ii) an engineering approach [59]. A more recent classification was described by Chikkerur et al. [62] which separated picture metrics into perceptual (HVS) oriented and natural vision characteristics oriented metrics. Perceptual oriented metrics utilize a visual modeling approach, and can be divided again into two categories: (i) a pixel domain approach and (ii) a multi-scale approach. Conversely, natural visual characteristics oriented metrics apply an engineering approach. Instead of using fundamental vision modeling, natural visual characteristics oriented metrics are based predominantly on extracting and analyzing certain features or artifacts in the video [58]. According to Chikkerur [62], they are divided into natural visual statistics and natural visual features based methods.

Human Visual System Modelling Oriented Metrics

Introducing HVS mechanisms into a quality metrics promotes a better correlation between subjective and objective video quality evaluation. However, the HVS is more

complex, so quality metrics typically incorporate only the most important HVS characteristics. Therefore, a visual modeling approach attempts to generate a better prediction of the perceived video quality by modeling the various features of human vision. For example, the human eye detects video contrast through the relative variation of luminance, so a quality model imitates this sensitivity through a spatiotemporal contrast sensitivity function (CSF) [63]. Quality evaluation procedures frequently employ the use of a mechanism that masks properties of the image or video content. When distortion appears in textured regions of an image as opposed to smooth regions, distortion visibility decreases. High levels of movement in a video also decrease the visibility of impairments.

Several of the objective image and video quality assessment approaches that are described in the literature employ a common error sensitivity-based philosophy [59]. The objective of this philosophy is to quantify the strength of the errors between the reference and the distorted signals in a perceptually meaningful way. In the first step of the assessment, various pre-processing procedures are implemented, including registration, color space transformation, light adaptation, and calibration for display devices. The second step focuses on channel decomposition which can be achieved by filtering or using a transform such as the discrete wavelet transform (DWT) or the discrete cosine transform (DCT). Following these transform processes, a CSF is implemented in order to estimate frequency responses of the HVS. After this, error normalization and masking are applied. Typically during this step the cross-channel masking of a visual channel by the contents of another is neglected. Lastly, this technique combines error signals from different channels into a single distortion value. To accomplish this, spatial pooling and temporal pooling across each video frame are employed. Further information about the error sensitivity-based framework can be seen in Wang, et al. [59] and Wang, et al. [64].

To quantify the visual integrity of natural images, Chandler and Hemami [65] describe a wavelet-based visual signal-to-noise ratio (VSNR). The VSNR is derived from near-threshold and supra-threshold properties of human vision. The complete VSNR calculation procedure is provided in [65]. In order to perform a visual quality assessment (VQA) application, as demonstrated in [62], the VSNR is applied frame-by-frame on the luminance component of the video. The resulting overall VSNR index of the video is calculated as the average of the frame level VSNR scores. Even though the authors of VSNR had not proposed such an extension, the same VSNR score calculation method was applied to videos in the present experiments.

Recently, Seshadrinathan and Bovik [66] [67] generated a MOtion-based Video Integrity Evaluation (MOVIE) index. Essentially, the MOVIE is a FR VQA algorithm that incorporates both spatial and temporal aspects of distortion assessment. To accomplish this, it utilizes a spatio-temporally localized, multi-scale decomposition of the reference and test videos using a family of spatio-temporal Gabor filters (three scales with 35 filters at each scale). Thus, MOVIE consists of two components: spatial MOVIE (SMOVIE), and temporal MOVIE (TMOVIE). The spatial MOVIE index quantifies spatial distortions in the video, whereas temporal MOVIE measures temporal distortions in the video by computing and using motion information from the reference video explicitly.

An objective video quality metric called foveated mean square error (FMSE) was reported by Rimac-Drlje, et al. [68]. The FMSE is founded on the MSE approach and assigns different weighted values to the errors at different points on the frame due to the assumption that the human eye focuses on a pixel at the center of the frame. The authors accounted for the decline in human eye contrast sensitivity when eccentricity on the retina rises and when retinal image velocity rises. Additionally, they implemented the

foveation-based contrast sensitivity to acquire foveation-based sensitivity for scenes with motion.

In a framework proposed by Barkowsky et al. [69], the authors added temporal distortion awareness to standard VQA algorithms. This technique utilizes motion estimation to track image areas over time. In order to evaluate the appearance of new image areas and the display time of objects, motion vectors and the motion prediction error are determined and assessed. Furthermore, this framework allows for a more exact judgment of degradations that attach to and remain with moving objects.

Zhao et al. [70] presented an innovative FR video quality metric called perceptual quality index (PQI). In this metric, multiple visual properties are utilized to mimic subjective evaluation on impaired videos. So as to detect and quantify perceptible distortions in both spatial and temporal channels, PQI employs visual performance information in foveal and extra-foveal vision and a spatial-temporal just noticeable difference (JND) model. In each channel, visual errors are summed and quality degradation over time is collected in order to model the visual persistence and recency effects. A final perceptual quality score is computed by transforming and fusing the intensities of spatial and temporal noises into quality scales.

Engineering Approach Oriented Metrics

The engineering approach to video quality metrics is characterized by the extraction and analysis of certain features such as structural elements and artifacts such as blockiness or blur in the video. Engineering metrics, also described as the top-down approach, evaluate the strength of video features in order to approximate the overall quality. Although the

engineering approach differs from fundamental vision modeling, it does not neglect the HVS, as it often accounts for psychophysical vision effects. The main distinction of the engineering approach is that its conceptual foundation lies in image content and distortion analysis.

To assess image quality, Wang et al. [71] designed a structural similarity (SSIM) index, that uses structural distortion as an estimate of the perceived visual distortion. In order to determine structural distortion, SSIM utilizes means, variances, and the covariance of original and distorted sequences. The benefit of SSIM indices is that they are calculated only for properly selected blocks instead of the whole frame, which lessens computational costs while still providing reliable experimental results.

Different extensions of the SSIM indices are designed for still images and then extended to the video. An example is the multi-scale SSIM (MS-SSIM) index for still images described by Zhou, et al. [72]. The MS-SSIM considers the dependence of image detail perceivability on sampling density of the image signal, the distance from the image plane to the observer, and the perceptual capacity of the observers visual system. In order to apply MS-SSIM to VQA, the MS-SSIM is implemented in the luminance component of each frame of the video. To obtain the overall MS-SSIM index for the video, the individual frame level quality scores are averaged.

Speed SSIM index is an additional VQA extension of the SSIM paradigm. This application pairs SSIM with statistical models of visual speed perception, as proposed by Zhou and Li [73]. In any particular video sequence speed, SSIM accounts for three types of motion fields (i) absolute motion, (ii) background motion, and (iii) relative motion, and a model of human visual speed perception [74]. This results in a calculation of perceived video quality.

Finally, Mittal et al. [75] propose a natural scene statistic-based distortion-generic blind/no-reference image quality assessment (IQA) model that operates in the spatial domain. Their model (called blind/referenceless image spatial quality evaluator (BRISQUE)) does not compute distortion-specific features, such as ringing, blur, or blocking, but instead uses scene statistics of locally normalized luminance coefficients to quantify possible losses of naturalness in the image due to the presence of distortions, thereby leading to a measure of quality.

5.3.3 Packet and Bitstream−based Metrics

The delivery of video over IP networks has been provided in a growing number of services (specifically IPTV). Therefore, the development of VQA metrics that consider the impact of network losses on video quality became imperative. To do this, it is important to determine the amount of video information lost in some packets in their transmission through the network. With little or no decoding, packet and bitstream-based metrics extricate certain parameters from the transport stream and the bitstream. As compared to the metrics that utilize fully decoded video, the packet and bitstream-based metrics result in lower bandwidth and processing requirements. One of these metrics is the V-Factor described by Winkler [76], and other examples are included by Versheure, et al. [77]. However, packet and bitstream-based metrics are limited because they are tailored to specific codecs and network protocols. Details about quality of sevice (QoS) and quality of experience (QoE) techniques for IPTV are included for instance by Krej [78].

5.4 Tracker Quality Assessment

The challenge now is to connect the methods of VQA with the multiple target tracking techniques using video. The chosen techniques must assess the video sequences at the pixel level where the tracking is performed and measurements are obtained.

Given the big variety of VQA methods introduced in the previous sections, finding the appropriate technique to use as part of our tracker quality assessment framework is a delicate task, given that our purpose is to use the information content at the pixel level and it is also important to use techniques that gives us consistent results and are valid from an engineering point of view. First then it is important to note that we can not use the techniques that use as a base a model of the HVS since tracking techniques have no similarity with our visual system. We can also rule out quality measures over networks or bitstream−based metrics since we are assessing the available video for the detection algorithms without having to look at what happens at his origin. Finally, simple data metrics such as MSE and PSNR are too general and do not look at the specific pixel locations.

In light of this, the best choice of methods to include on a framework for tracker quality assessment comes from the engineering approach to visual quality assessment, more specifically methods that look at the quality of the pixel on different scales, since detectors for objects on video use this information, where the main connection between tracking on videos and video quality could be encountered. Given all the mentioned characteristics, we use widely available VQA engineering approaches: BRISQUE and MS-SSIM.

5.4.1 Framework Description

The framework is simple and operates in the manner of usual image filtering techniques. Performing sequential spatial filtering using BRISQUE to weight each pixel and the MTT transform to weight the pixels that are in the areas where measurements are obtained. We also make use of the MS-SSIM as a comparison of the images but not on the usual manner of a reference image but to compare information encoding evolution in time, inspired on the work by Wang et al. [73].

The process of incorporating the MTT transform into a quality assessment framework is divided into three steps (Figure. 5.1), in order to incorporate the concepts from the marginalized transform and the visual quality assessment. First, each frame is obtained and weighted by the quality score obtained by BRISQUE, parallel to this the detector that is going to be used for the tracking application is used on each frame, obtaining real measurements. The BRISQUE weighting has an objective the normalization of images in order to eliminated any biases caused for high or low quality videos. The obtained measurements are used to numerically calculate the MTT transform for each technique, in order to do this each measurement is assumed to represent a target with all the probabilities provided by the technique's assumptions. On a second step the BRISQUE weighted image is changed in by applying the weighting obtain during the transform evaluation, but only in the pixel regions where measurement were obtained. Finally the MS-SSIM is used to compare the weighted images from one frame to the next (as shown in Figure 5.1). The tracker quality assessment (TQA) is the mean of the MS-SSIM over time. In practice, for the usual application of MS-SSIM the higher its values is the higher the quality of an image is, since the reference is the image with the perfect quality.

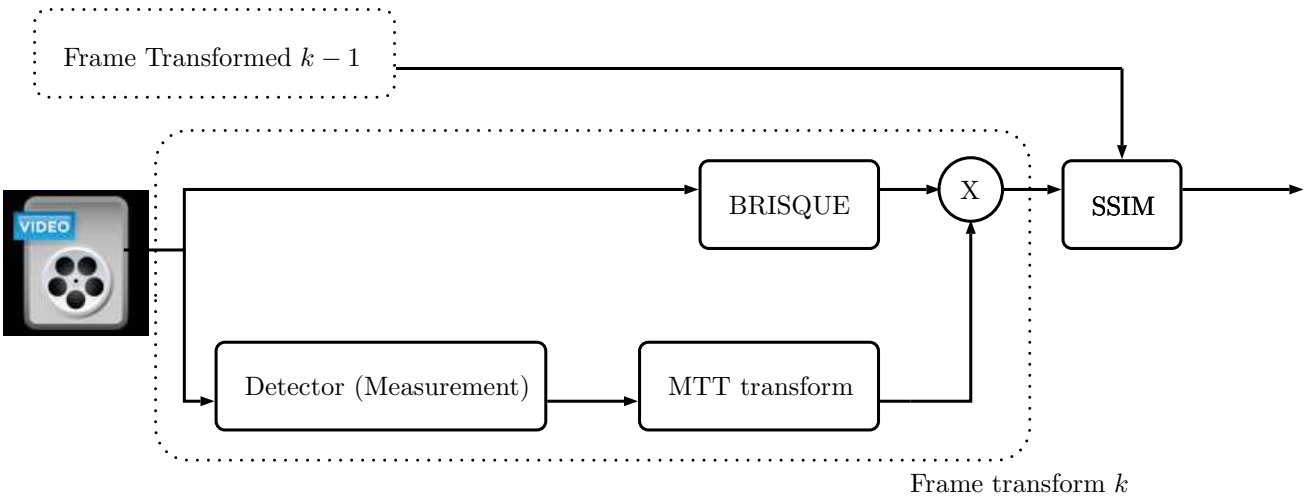


Figure 5.1: Framework for the application of the MTT marginalized transform combined with visual quality assessment

CHAPTER 6

Implementations and Results

This chapter is divided in two main parts. First we introduce the way in which the marginalized transforms are calculated in order to be used as part of the TQA framework. Finally we present the results for a variety of experiments exploring possible applications.

6.1 Tracker Quality Index Implementations

The implementation of the TQA framework needs the evaluation of each of the transform expressions presented in Table 4.1. This might seem like a simple task, but it needs special care in order to maintain the assumptions for the techniques and more importantly the actual implementations. The main assumption for the transform evaluation is that a target is present where a measurement is present (marginalization) and a transform value is calculated for each of them, then for each frame, if we have m measurements that give us m transform values that are added to find the total value for that set of measurements.

6.1.1 Joint Probabilistic Data Association

Calculating the transform for the JPDA has the same form than the presented in the transform Table, basically the only consideration to be taking in account is making sure that scaling remains proper for each measurement.

Algorithm 1 JPDA Marginalized transform calculation

while video is running **do**
  $\boldsymbol{Y} \leftarrow$ set of measurements for current frame ▷ $\boldsymbol{Y}$ is a matrix of $l \times m$
  $\Psi = 0$
  for $i = 1$ to m **do**
    $\boldsymbol{x} = \boldsymbol{Y}_i$
    ▷ assuming a target is present at each measurement
    $\boldsymbol{x}_+ = A \cdot \boldsymbol{x}$
    $Yest = H \cdot \boldsymbol{x}_+$
    $P_+ = A \cdot P_0 \cdot A^T + Q$
    $S = H \cdot P_+ \cdot H^T + R$
    $BMD = a_x \cdot \mathcal{N}(\boldsymbol{x}|\boldsymbol{x}_+, P_+) + b_x \cdot \mathcal{N}(Y_i|Y_+, S)$
    ▷ This accounts for detection and model effects
    $Clutter = e^{-\frac{1}{V}+c_d}$
    ▷ Effects of clutter, c_d is the parameter we change for evaluation
    $\Psi = \Psi + k * Clutter * BMD^n$
    ▷ Where n is the previously known number of targets, and k is a normalizing constant
  end for
end while

6.1.2 Multiple Hypothesis Tracking

For the MHT we need to introduce the expression for the calculation of the coefficients γ and χ. Using the gaussian assumption and the expressions for hypothesis evaluation from the MHT we have

$$\chi_{MHT} = \frac{1}{c} \cdot \frac{\nu! F!}{\bar{Q}!} \cdot e^{-p_b} \cdot p_d \cdot \left(e^{-c_d}\right)^F \tag{6.1.1}$$

$$\gamma_{MHT} = \frac{1}{c} \cdot \frac{\nu! F!}{\bar{Q}!} \cdot e^{-c_d} \cdot \left(e^{-p_b}\right)^\nu \tag{6.1.2}$$

where F is the number of false alarms, ν is the number of new targets, $\bar{Q}$ is the number of targets, and again p_b and p_d are the probabilities of birth and death, respectively [37] [79]. All the former quantities are calculated using random sampling according to the appropriate distribution: binomial distribution for ν, and Poisson distribution for F. $\bar{Q}$ is approximated by the number of measurements for each frame. It is important to remember that in this

case the probability of detection and the innovation probability density function are already taken into account inside the Bayes-Markov filter portion of the marginalized transform expression. In general is shares the core algorithm with the MCMC (see Algorithm 2).

6.1.3 Markov Chain Monte Carlo

For the MCMC the evaluation has a different nature, since it depends on the moves mentioned in Section 3.5 and the implementation selected. In this case we have

$$\chi_{MCMC} = (1 - p_b) \cdot p_d \cdot \tau \cdot C \tag{6.1.3}$$

$$\gamma_{MCMC} = p_b \cdot (1 - p_d) \cdot C \tag{6.1.4}$$

where p_b is the probability of birth, p_d is the probability of death, τ is the target prior, and C is the clutter prior for a target. The value for each of these variables is calculated by performing a small Metropolis-Hastings sampling [28] using the different assumptions present in the implementation. To calculate those values we used the same criteria presented by by Särkkä *et al.* in [43]. The value of C depends on sampling a Poisson distribution with density c_d and it is equals the inverse of the surveillance volume if the target is said to be a false alarm, otherwise it is one. τ depends on sampling from a given target representing a false alarm or existing target, and it is obtained by sampling a Poisson distribution with intensity p_b and assigning the value of one minus the inverse surveillance volume.

Algorithm 2 MCMC and MHT Marginalized transform calculation.

while video is running **do**
  $\boldsymbol{Y} \leftarrow$ set of measurements for current frame
    ▷ $\boldsymbol{Y}$ is an $l \times m$ matrix
  $\Psi = 0$
  for $i = 1$ to m **do**
    $\boldsymbol{x} = \boldsymbol{Y}_i$ ▷ Assume a target is present at each measurement
    $\boldsymbol{x}_+ = A \cdot \boldsymbol{x}$
    $Y_+ = H \cdot \boldsymbol{x}_+$
    $P_+ = A \cdot P_0 \cdot A^T + Q$
    $S = H \cdot P_+ \cdot H^T + R$
    $BMD = a_x \cdot \mathcal{N}(\boldsymbol{x}|\boldsymbol{x}_+, P_+) + b_x \cdot \mathcal{N}(Y_i|Y_+, S)$
      ▷ Account for detection and model effects
    $S_{data} = H \cdot P_{birth} \cdot H^T + R$
    $BMD_{data} = a_x \cdot \mathcal{N}(x_+|M_{birth}, P_{birth}) + b_x \cdot \mathcal{N}(Y_+|Y_i, S_{data})$
      ▷ Effects of birth density with a Normal distribution
      ▷ with mean M_{birth} and covariance P_{birth}
    Evaluate χ according to Eq. (6.1.1) or Eq. (6.1.3).
    Evaluate γ according to Eq. (6.1.2) or Eq. (6.1.4).
    $\Psi = \Psi + (1 - \chi + \chi * BMD)^m \cdot (1 - \gamma + \gamma * BMD_{data})^m$
  end for
end while

6.1.4 Probability Hypothesis Density

This implementation does not require the calculation of extra coefficients as in the former cases, and in consists on evaluating simply on evaluating the representation in the transform Table, but expanded according to Eq. (71) in [53], but evaluated according to the marginalization concept. Given that it is an exponential equation it can presents really small numerical results, thus we can use the logarithmic scale and do the appropriate scaling.

6.2 Experimental Evaluation

On its simplest form the evaluation of the tracker quality assessment is straightforward. Given a video sequence we can simply apply the tracker quality assessment framework and obtain a quantity that predicts the expected performance of the technique.

Algorithm 3 PHD Marginalized transform calculation

while video is running **do**
  $\boldsymbol{Y} \leftarrow$ set of measurements for current frame
    ▷ $\boldsymbol{Y}$ is an $l \times m$ matrix
  $\Psi = 0$
  for $i = 1$ to m **do**
    $\boldsymbol{x} = \boldsymbol{Y}_i$ ▷ Assume a target is present at each measurement
    $\boldsymbol{x}_+ = A \cdot \boldsymbol{x}$
    $Y_+ = H \cdot \boldsymbol{x}_+$
    $P_+ = A \cdot P_0 \cdot A^T + Q$
    $S = H \cdot P_+ \cdot H^T + R$
    $BMD = a_x \cdot \mathcal{N}(\boldsymbol{x}|\boldsymbol{x}_+, P_+) + b_x \cdot \mathcal{N}(Y_i|Y_+, S)$
      ▷ Account for detection and model effects
    $PHD = e^{\left(-\frac{1}{V} - m + c_d + m * BMD\right)}$
      ▷ Account for detection and model effects
    $\Psi = \Psi + k * log\,(PHD)$ ▷ k for appropriate scaling
  end for
end while

In order to accomplish this experiment we need to run the tracking techniques on the video sequence to have a real performance measure and compare with the prediction, it is really important to note that the actual performance is not important in our case, but the relative prediction of the performance. For this work we use the optimal subpattern assignment (OSPA) metric [8] since it has been widely used as one of the main performance metrics for non-labeled multiple target tracking applications[1]. Finally, the base implementations used in this work for the MHT comes from the work of Antunes et al. [40], the MCMC data association and JPDA from the toolbox by Särkkä et al. [43], and the PHD comes from the implementation by Bryan Clarke of [47], all extended or modified to carry out centroid tracking on videos with a constant velocity model [5].

The video sequences used are the widely used and publicly available VSPETS 2003 INMOVE soccer dataset[2], using a red detector to obtain the centroids of one of the teams (Liverpool), which in general provides very accurate measurements, and the 2009

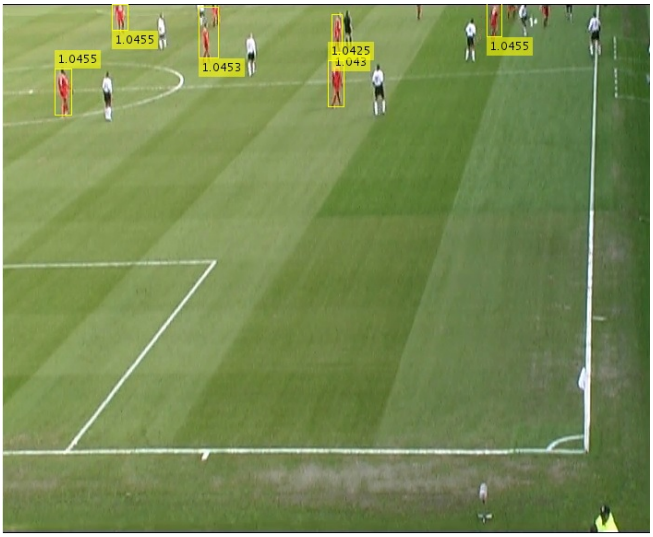
[1] All tracking scenarios are performed 15 times for Monte Carlo Simulations
[2] The dataset can be found in `ftp://ftp.cs.rdg.ac.uk/pub/VS-PETS/`

BAHNHOF sequence which is a moving camera scenario[3], where the targets are pedestrians and the detections were made with an HOG and the clutter and accuracy of the detection is not as high. The main assumption for the transform evaluation is that a target is present where a measurement is present (at marginalization) and a transform value is calculated for each of them, then for each frame, if there are q measurements we obtain q transform values that are superposed since the measurement space is unique. This also indicates that the values for m and n in the transform Table 4.1 are equal to the number of measurements obtained at each frame.

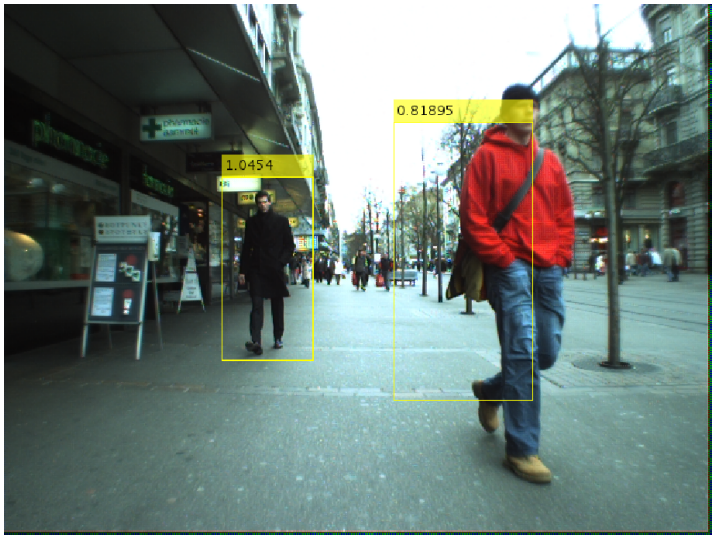
The tracker quality assessment framework was used to evaluate each tracker using different sets of basic assumptions, such as varying the false alarm intensities, the probability of detection and/or probability of birth. Although every parameter could be changed for comparison or tuning, from the model and measurement covariance, to the birth densities or the motion model itself, for simplicity and conciseness we limit our evaluation to a limited amount of parameters that present good performance variation. For the different scenarios, after extensive experimentation it was found that changing the false alarm density affects this specific MHT implementation the most, and hence that was the parameter chosen for evaluation, the same can be said about the JPDA and PHD implementations. In the case of the MCMC technique changing only one parameter does not affect the performance given the way in which the MCMC explores the hypothesis space. Table 6.1 presents the parameter values used in our evaluation. In the table, c_d is the clutter or false alarm density, p_d is the probability of death, and p_b is the probability of birth.

We analyze three sets of values that can confirm or disproof the validity of our hypothesis. Total OSPA values are provided to show the real performance of the techniques

[3]The dataset can be found in `https://data.vision.ee.ethz.ch/cvl/aess/dataset/`



(a) Soccer scenario VSPETS 2003 INMOVE



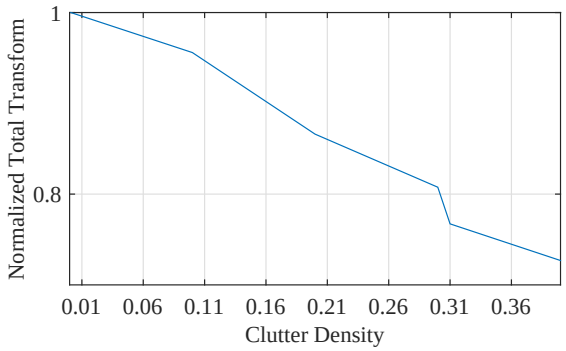
(b) Moving camera scenario 2009 BAHNHOF

Figure 6.1: Snapshot of the datasets analysed

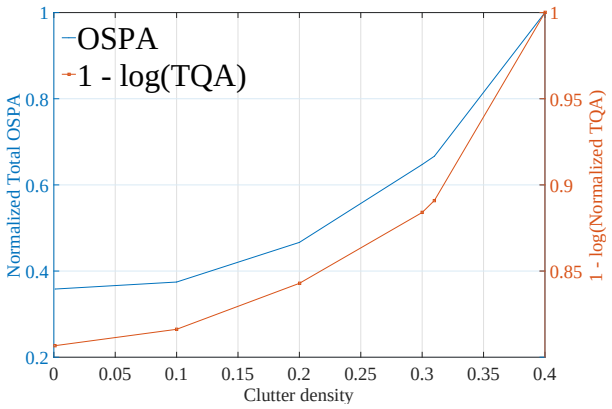
Table 6.1: Sets of assumption for the MCMC in the soccer scenario

MCMC Assumption set			
1	$c_d = 1/1000$	$p_d = 0.547$	$p_b = 0.1$
2	$c_d = 1/240$	$p_d = 0.8$	$p_b = 0.1$
3	$c_d = 1/1000$	$p_d = 0.9$	$p_b = 0.1$
4	$c_d = 1/100$	$p_d = 0.9$	$p_b = 0.1$
5	$c_d = 1/3$	$p_d = 0.547$	$p_b = 0.8$

using the ground truth, and consists on adding the frame by frame OSPA value. Total MTT transform represents the aggregated value of the transform evaluated for each frame. Finally, the TQA represents the output of the framework presented in Chapter 5. All quantities have been normalized by their largest value in order to facilitate visualization, including the $1 - log(NormalizedTQA)$ in order to give a better comparison with the tracker performance.



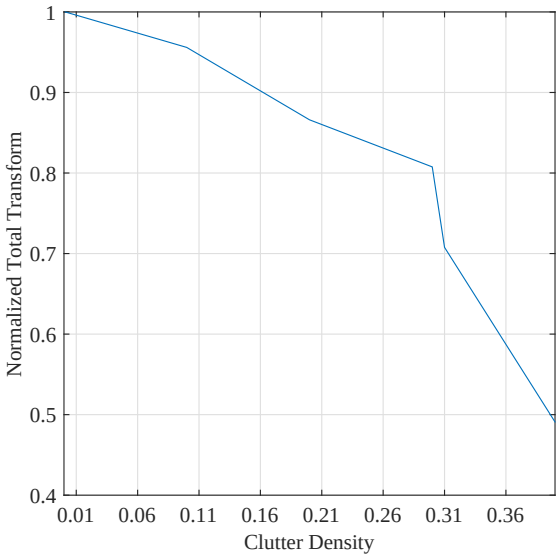
(a)



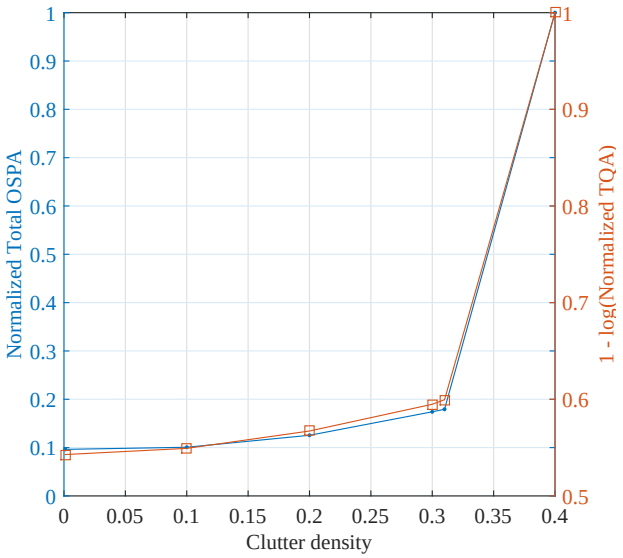
(b)

Figure 6.2: JPDA results for soccer scenario

It is important to consider each scenario separately and for each technique. In the case concerning the soccer scenario it can be observed that the TQA does an excellent job predicting the performance for the MHT technique as can be seen in Figure 6.3.b, given that the smaller the OSPA the better the overall performance. In this case the total MTT transform also performs a good job (Figure 6.3.a), but it is important to remember that it



(a)



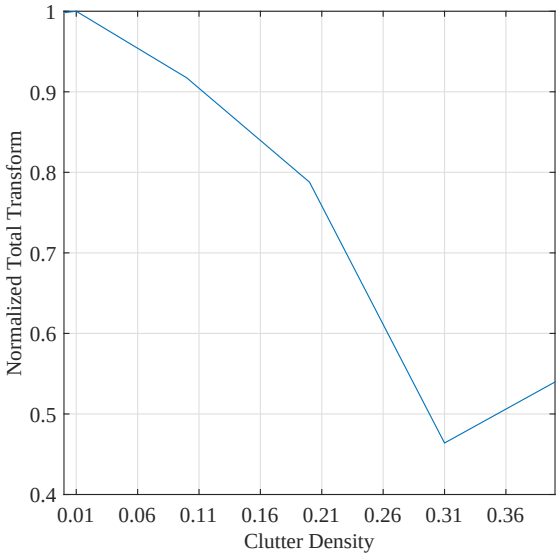
(b)

Figure 6.3: MHT results for soccer scenario

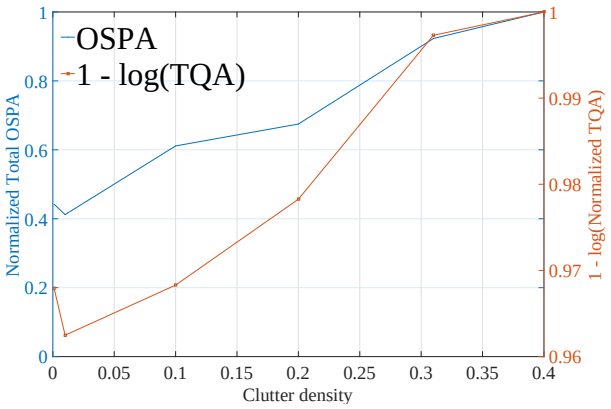
only takes in account the detection and not the video quality. For the MCMC we can observe small variation on the performance prediction for the first three sets of assumptions which is reflected on the actual OSPA in Figure 6.7. In the case of the JPDA we can only analyse the soccer scenario since we know how many targets are present during the length of the video, and we can observe a similar effect on the performance when changing the values

of the clutter density, and the performance prediction results show great concordance with the OSPA variations (Figure 6.2.b). Finally the PHD filter follows the observed pattern so far, presenting a really good prediction over the clutter density spectrum studied (see Figure 6.5.b). In general for this scenario the TQA framework performs really well, this is mostly due to the high quality of the measurements and the reduced number of false alarms and video changes, given the stationary camera.

The moving camera scenario and the different detector present more challenges for

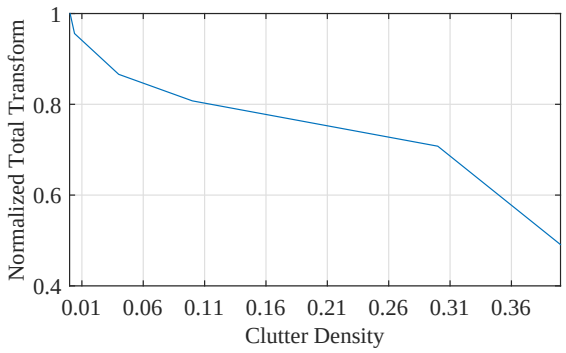


(a)

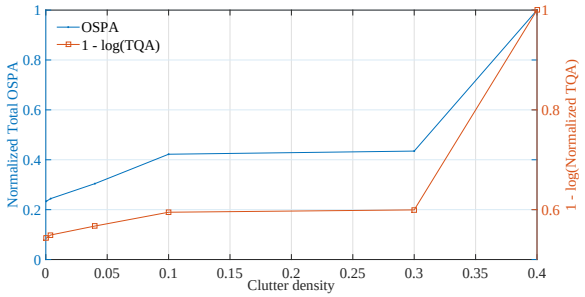


(b)

Figure 6.4: MHT results for moving camera scenario



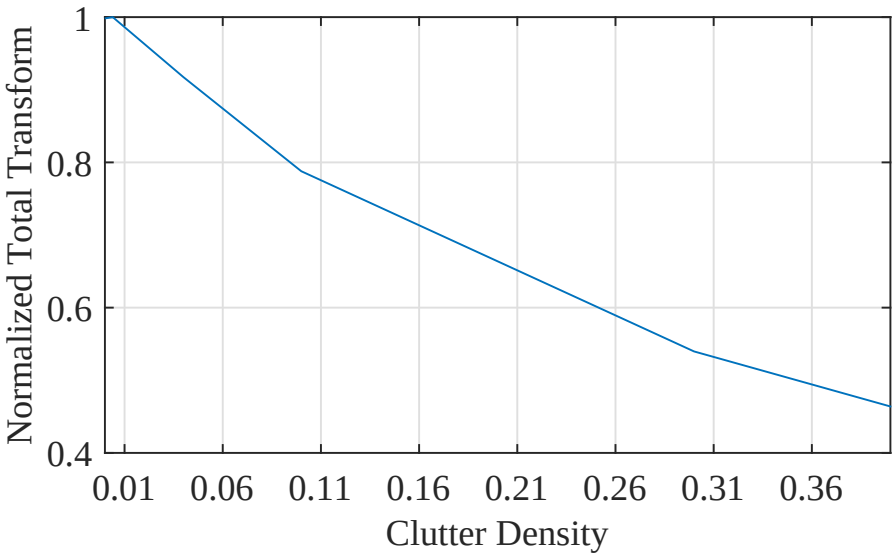
(a)



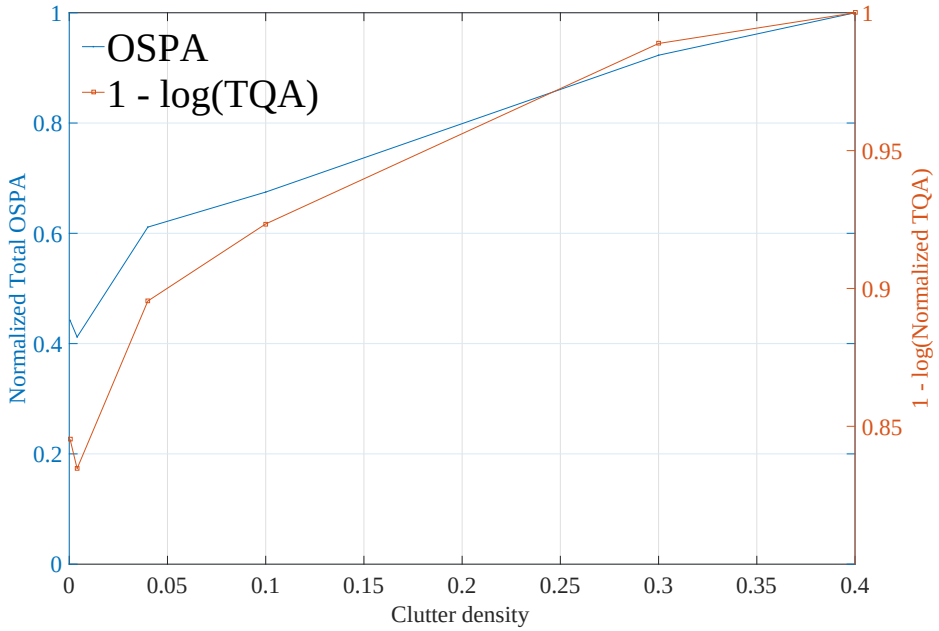
(b)

Figure 6.5: PHD results for soccer scenario

the evaluated tracking techniques and it has the same effect on the TQA framework. The measurements were corrected given the world coordinates of the camera that are included in the dataset, and technically this accounts for the camera motion in order to make more accurate predictions. For the MCMC it can be observed that the variation of the TQA value is much smaller than the actual OSPA variation but the trends are as precise as in the former scenario when accounting for camera motion (Figures 6.4.b 6.6.b, and 6.8.b and c) with the transform presenting a really good measure in this case too, since it is not as affected by the bigger frame to frame change change in information produced by the motion (Figures 6.4.a and 6.8.a). The scale change for the framework is usually small (see 6.4.b and 6.6.b for instance), this tells us that the TQA prediction should be considered more of a relative than an absolute value for comparison or tuning purposes.

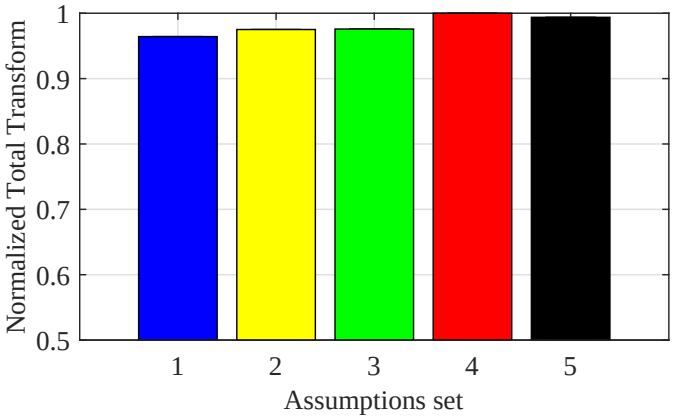


(a)

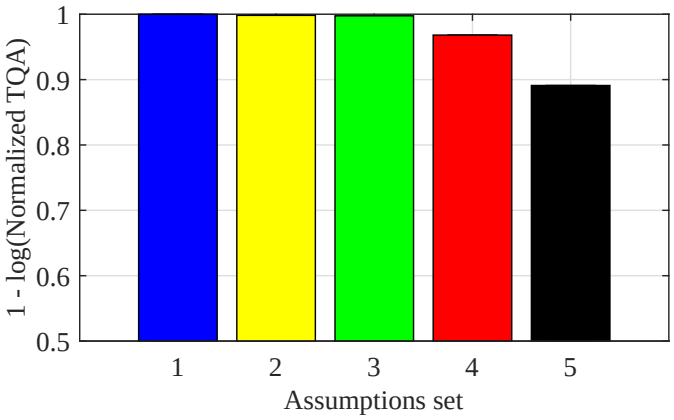


(b)

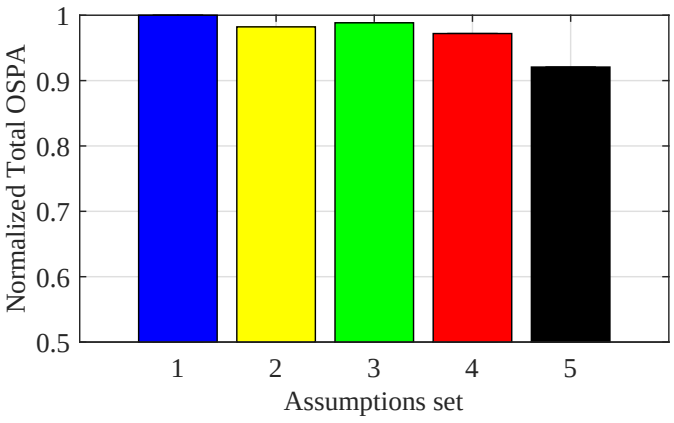
Figure 6.6: PHD results for moving camera scenario



(a)

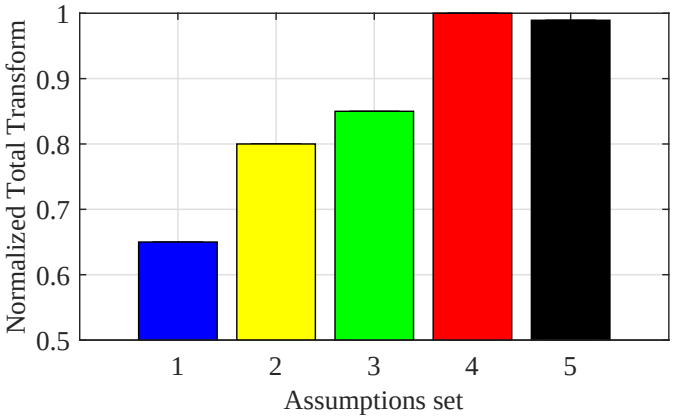


(b)

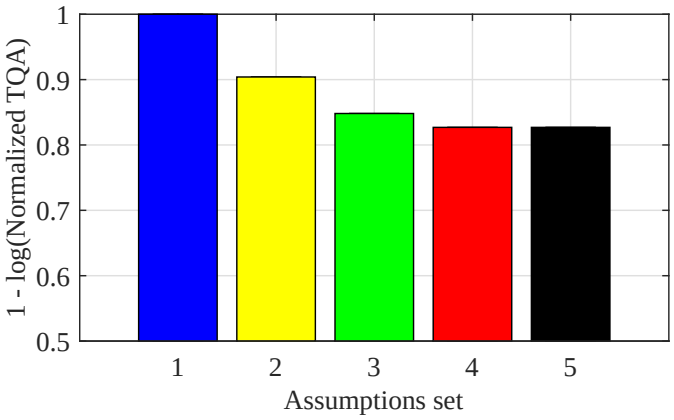


(c)

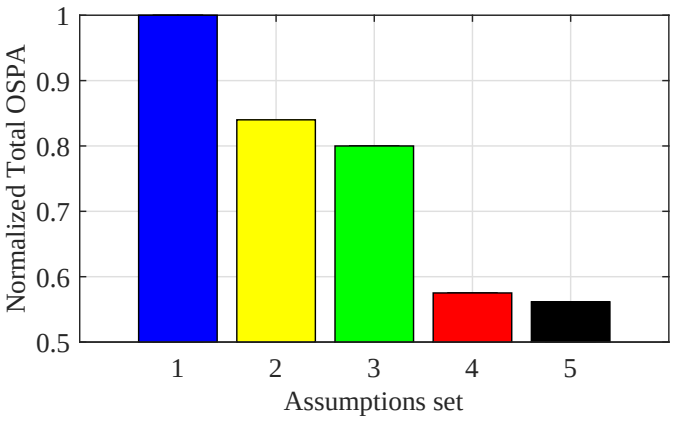
Figure 6.7: MCMC results for soccer scenario



(a)



(b)



(c)

Figure 6.8: MCMC results for moving camera scenario

CHAPTER 7

Conclusions and Recommendations

7.1 Conclusions

The mathematical framework given by finite point processes allows for the introduction of novel concepts that can be used for a compact representation of MTT techniques that can be useful to obtain more information about the nature of these techniques and perform application beyond pure target tracking. We presented a new framework that allow us to predict the performance of MTT techniques taking advantage of these concepts without performing tracking. This has not been done before and presents a completely novel application with theoretical and practical implications.

The MTT transform gives us an insight on how the different assumptions of a MTT technique affect the way in which the likelihood uses the information content of the measurements, which can be clearly observed in the results for the different techniques and scenarios. The MTT transform by itself can only gives us information about the effective use of the measurements but not a complete prediction since the scenario in which the tracking is occurring affects the performance.

Visual quality assessment techniques can be successfully integrated with the proposed transform to give a more realistic performance prediction that takes in account the problems of quality present in the video sequences. The performance of the framework is not perfect as was observed but in general it will tend to show the actual performance prediction. It is

still important to have a good body of knowledge of target tracking in order to use this tool, but it can be prepared as a general design tool.

7.2 Future Work and Recommendations

In order to make this results more definitive for the application on video sequences it would be important to consider more tracking techniques and tracking scenarios, including moving cameras, different detectors and motion models. Other important aspect to take in account is the use or construction of different quality metrics that serve a more focus purposes, for example changing the way in which the reference comparison are computed. In terms of computer vision applications, there is one more possible use for this framework in combination with optimization techniques, in order to conduct a search to find the tuning parameters for the different techniques.

A second branch of future work includes looking at other scenarios where a similar framework can be defined, expanding the possibilities of application outside computer vision. This touches in a more complex subject, since obtaining good evaluations or quality assessment of physical systems such as radar or robotics platforms has not been explored in the past.

References

[1] G. W. Pulford, "Taxonomy of multiple target tracking methods", *IEEE Proceedings on Radar, Sonar and Navigation*, vol. 152, no. 5, pp. 291–304, 2005, ID: 1.

[2] Ronald PS Mahler, *Statistical multisource-multitarget information fusion*, Artech House, Inc., 2007.

[3] Roy L. Streit, *Poisson Point Processes: Imaging, Tracking, and Sensing*, Springer Science and Business Media, 2010.

[4] Muralidhar Krishna Yeddanapudi, "Estimation and data association algorithms for multisensor-multitarget tracking", *ProQuest Dissertations and Theses*, 1996.

[5] Emilio Maggio and Andrea Cavallaro, *Video tracking: theory and practice*, John Wiley and Sons, 2011.

[6] E. Loutas, N. Nikolaidis, and I. Pitas, "Evaluation of tracking reliability metrics based on information theory and normalized correlation", in *Proceedings of the 17th International Conference on Pattern Recognition ICPR*, 2004, vol. 4, pp. 653–656 Vol.4.

[7] K. Kao Edward, P. Daggett Matthew, and B. Hurley Michael, "An information theoretic approach for tracker performance evaluation", in *IEEE 12th International Conference on Computer Vision*, 2009, pp. 1523–1529.

[8] Branko Ristic, Ba-Ngu Vo, Daniel Clark, and Ba-Tuong Vo, "A metric for performance evaluation of multi-target tracking algorithms", *IEEE Transactions on Signal Processing*, vol. 59, no. 7, pp. 3452–3457, 2011.

[9] Keni Bernardin and Rainer Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics", *Journal on Image and Video Processing*, vol. 2008, pp. 1, 2008.

[10] Kevin Smith, Daniel Gatica-Perez, Jean-Marc Odobez, and Sileye Ba, "Evaluating multi-object tracking", in *Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, CVPR Workshops*. 2005, pp. 36–36, IEEE.

[11] Abdullahi Daniyan, "Performance analysis of sequential monte carlo mcmc and phd filters on multi-target tracking in video", in *Proceedings of the 2014 European Modelling Symposium*. 2014, pp. 195–202, IEEE Computer Society.

[12] Loris Bazzani, Domenico Bloisi, and Vittorio Murino, "A comparison of multi hypothesis kalman filter and particle filter for multi-target tracking", in *Performance Evaluation of Tracking and Surveillance workshop at CVPR*, 2009, pp. 47–54.

[13] Laura Leal-Taixá, Anton Milan, Ian Reid, Stefan Roth, and Konrad Schindler, "Motchallenge 2015: Towards a benchmark for multi-target tracking", *arXiv preprint*, 2015.

[14] Y. Ho and R. Lee, "A bayesian approach to problems in stochastic estimation and control", *Automatic Control, IEEE Transactions on*, vol. 9, no. 4, pp. 333 – 339, oct 1964.

[15] Andrew H. Jazwinski, *Stochastic Processes and Filtering Theory*, Academic Press, apr

1970.

[16] R L Stratonovich, *Conditional Markov Processes And Their Application To The Theory Of Optimal Control*, American Elsevier Publ CO, 1968.

[17] Rudolph Emil Kalman, "A new approach to linear filtering and prediction problems", *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[18] H. W. Sorenson and D. L. Alspach, "Recursive bayesian estimation using gaussian sums", *Automatica*, vol. 7, no. 4, pp. 465–479, jul 1971.

[19] Genshiro Kitagawa, "Non-Gaussian State-Space Modeling of Nonstationary Time Series", *Journal of the American Statistical Association*, vol. 82, no. 400, pp. 1032–1041, 1987.

[20] J. Handschin, "Monte Carlo Techniques for Prediction and Filtering of Non-Linear Stochastic Processes", *Automatica*, vol. 6, no. 4, pp. 555–563, jul 1970.

[21] D. Q. Mayne and J. E. Handschin, "Monte carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering", *International Journal of Control*, vol. 5, no. 5, pp. 547–559, 1969.

[22] N.J. Gordon, D.J. Salmond, and A.F.M. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation", *Radar and Signal Processing, IEEE Proceedings*, vol. 140, no. 2, pp. 107–113, apr 1993.

[23] Athanasios Papoulis, *Probability, random variables, and stochastic processes*, McGraw-Hill, Boston, 2002.

[24] Anton Haug, *Bayesian Estimation and Tracking a Practical Guide*, John Wiley and Sons, Hoboken, 2012.

[25] Simo Srkk, "Recursive bayesian inference on stochastic differential equations", Tech. Rep., Helsinki University of Technology, 2006.

[26] R.H. Bishop and A.C. Antoulas, "Nonlinear approach to aircraft tracking problem", *AIAA Journal of Guidance, Control, and Dynamics*, vol. 17, no. 5, pp. 1124–1130, 1994.

[27] Juan E. Tapiero and Robert H. Bishop, "Bayesian estimation for tracking of spiraling reentry vehicles", *AIAA guidance, navigation, and control (GNC) conference*, 2013.

[28] Dirk Kroese, *Handbook of Monte Carlo methods*, Wiley, Hoboken, N.J, 2011.

[29] Shaolin Lv, Jiabin Chen, and Zhide Liu, "Udut continuous-discrete unscented kalman filtering", in *Proceedings of the 2008 Second International Symposium on Intelligent Information Technology Application - Volume 02*, Washington, DC, USA, 2008, IITA '08, pp. 876–879, IEEE Computer Society.

[30] Jesper Carlsson, Kyoung-Sook Moon, Anders Szepessy, and Ral Tempone Georgios Zouraris, "Stochastic differential equations: Models and numerics", 2010.

[31] R. H. Bishop and A. C. Antoulas, "A nonlinear approach to the aircraft tracking problem", *AIAA Journal of Guidance, Control, and Dynamics*, vol. 17, no. 5, pp. 1124–1130, 1994.

[32] Samuel Blackman, *Design and analysis of modern tracking systems*, Artech House, Boston, 1999.

[33] Yaakov Bar-Shalom and Xiao-Rong Li, *Multitarget-multisensor tracking: principles and techniques*, vol. 19, YBS Storrs, Conn., 1995.

[34] Sudha Challa, *Fundamentals of object tracking*, Cambridge University Press, 2011.

[35] DE Tinne, "Rigorously bayesian multitarget tracking and localization", *Dissertation*, 2010.

[36] O. Dubois-Matra, *Development of Multisensor Fusion Techniques with Gating Networks Applied to Reentry Vehicles*, PhD thesis, University of Texas, Austin, may 2003.

[37] Donald B. Reid, "An algorithm for tracking multiple targets", *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.

[38] Samuel S. Blackman, "Multiple hypothesis tracking for multiple target tracking", *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5–18, 2004.

[39] Katta G. Murty, "An algorithm for ranking all the assignments in order of increasing cost", *Operations research*, vol. 16, no. 3, pp. 682–687, 1968.

[40] David Miguel Antunes, David Martins de Matos, and Jose Gaspar, "A library for implementing the multiple hypothesis tracking algorithm", *arXiv preprint*, 2011.

[41] Thomas E. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association", *IEEE Journal of Oceanic Engineering*, vol. 8, no. 3, pp. 173–184, 1983.

[42] KG Murthy, "An algorithm for ranking all the assignments in order of increasing costs", *Operations research*, vol. 16, no. 3, pp. 682–687, 1968.

[43] Jouni Hartikainen and Simo Särkkä, "RBMCDAbox-Matlab toolbox of rao-blackwellized data association particle filters", *documentation of RBMCDA Toolbox for Matlab V*, 2008.

[44] Songhwai Oh, Stuart Russell, and Shankar Sastry, "Markov chain monte carlo data association for general multiple-target tracking problems", in *43rd Conference on Decision and Control*. 2004, vol. 1, pp. 735–742, IEEE.

[45] Daniel Duckworth, *Monte carlo methods for multiple target tracking and parameter estimation*, PhD thesis, Citeseer, 2012.

[46] Melanie Anne Edith Bocquel, "Random finite sets in multi-target tracking: efficient sequential mcmc implementation", 2013.

[47] Ba-Ngu Vo and Wing-Kin Ma, "The gaussian mixture probability hypothesis density filter", *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4091–4104, 2006.

[48] Bharath Kalyan, "Unified random finite set theoretic approach to autonomous underwater vehicle navigation", *Dissertation*, 2011.

[49] Brunella Marta Spinelli, "Statistical inference for stable point processes", *Dissertation*, 2012.

[50] JE Moyal, "The general theory of stochastic population processes", *Acta mathematica*, vol. 108, no. 1, pp. 1–31, 1962.

[51] M. Westcott, "The probability generating functional", *Journal of the Australian Mathematical Society*, vol. 14, pp. 448–466, 1972.

[52] Roy Streit, "The probability generating functional for finite point processes, and its application to the comparison of phd and intensity filters", *Journal of Advances in Information Fusion*, 2013.

[53] Roy Streit, Christoph Degen, and Wolfgang Koch, "The pointillist family of multitarget tracking filters", *arXiv preprint*, 2015.

[54] Joshua Darr EmBree, "Spatial temporal exponential-family point processes for the evolution of social systems", 2015.

[55] Noel Cressie and GM Laslett, "Random set theory and problems of modeling", *SIAM Review*, vol. 29, no. 4, pp. 557–574, 1987.

[56] Roy Streit, "Saddle point method for jpda and related filters", in *Information Fusion (Fusion), 2015 18th International Conference on*. 2015, pp. 1680–1687, IEEE.

[57] Robin Pemantle and Mark C. Wilson, "Analytic combinatorics in several variables", *AMC*, vol. 10, pp. 12.

[58] Mario Vranjes, Snjeana Rimac-Drlje, and Kresimir Grgic, "Review of objective video quality metrics and performance comparison using different databases", *Signal Processing: Image Communication*, vol. 28, pp. 1–19, 1 2013.

[59] Zhou Wang, Hamid R. Sheikh, and Alan C. Bovik, "Objective video quality assessment", *The handbook of video databases: design and applications*, pp. 1041–1078, 2003.

[60] Muhammad Shahid, Andreas Rossholm, Benny Lvstrm, and Hans-Jrgen Zepernick, "No-reference image and video quality assessment: a classification and review of recent approaches", *EURASIP Journal on Image and Video Processing*, , no. 1, pp. 1–32, 2014.

[61] Stefan Winkler and Praveen Mohandas, "The evolution of video quality measurement: from psnr to hybrid metrics", *IEEE Transactions on Broadcasting*, vol. 54, no. 3, pp. 660–668, 2008.

[62] Shyamprasad Chikkerur, Vijay Sundaram, Martin Reisslein, and Lina J. Karam, "Objective video quality assessment methods: A classification, review, and performance comparison", *IEEE Transactions on Broadcasting*, vol. 57, no. 2, pp. 165–182, 2011.

[63] Peter GJ Barten, *Contrast sensitivity of the human eye and its effects on image quality*, vol. 72, SPIE press, 1999.

[64] Zhou Wang, Ligang Lu, and Alan C. Bovik, "Video quality assessment based on structural distortion measurement", *Signal Processing: Image Communication*, vol. 19, no. 2, pp. 121–132, 2004.

[65] D. M. Chandler and S. S. Hemami, "VSNR: A wavelet-based visual signal-to-noise ratio for natural images", *IEEE Transactions on Image Processing*, vol. 16, no. 9, pp. 2284–2298, 2007.

[66] Kalpana Seshadrinathan and Alan C. Bovik, "Motion-based perceptual quality assessment of video", in *ISandT-SPIE Electronic Imaging*. 2009, International Society for Optics and Photonics.

[67] Kalpana Seshadrinathan and Alan Conrad Bovik, "Motion tuned spatio-temporal quality assessment of natural videos", *IEEE Transactions on Image Processing*, vol. 19, no. 2, pp. 335–350, 2010.

[68] Snjezana Rimac-Drlje, Mario Vranjes, and Drago Zagar, "Foveated mean squared error, a novel video quality metric", *Multimedia Tools and Applications*, vol. 49, no. 3, pp. 425–445, 2010.

[69] Marcus Barkowsky, Jens Bialkowski, Bjrn Eskofier, Roland Bitto, and Andre Kaup, "Temporal trajectory aware video quality measure", *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 2, pp. 266–279, 2009.

[70] Yin Zhao, Lu Yu, Zhenzhong Chen, and Ce Zhu, "Video quality assessment based on measuring perceptual noise from spatial and temporal perspectives", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 12, pp. 1890–1902, 2011.

[71] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P. Simoncelli, "Image quality assessment: from error visibility to structural similarity", *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[72] Zhou Wang, Eero P. Simoncelli, and Alan C. Bovik, "Multiscale structural similarity for image quality assessment", 2003, vol. 2, pp. 1398–1402, IEEE.

[73] Zhou Wang and Qiang Li, "Video quality assessment using a statistical model of human visual speed perception", *JOSA*, vol. 24, no. 12, pp. B61–B69, 2007.

[74] Alan A. Stocker and Eero P. Simoncelli, "Noise characteristics and prior expectations in human visual speed perception", *Nature neuroscience*, vol. 9, no. 4, pp. 578–585, 2006.

[75] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik, "No-reference image quality assessment in the spatial domain", *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4695–4708, 2012.

[76] Stefan Winkler, *Digital video quality: vision models and metrics*, John Wiley and Sons, 2005.

[77] Oliver Verscheure, Pascal Frossard, and Maber Hamdi, "User-oriented QoS analysis in MPEG-2 video delivery", *Real-Time Imaging*, vol. 5, no. 5, pp. 305–314, 1999.

[78] Jaroslav Krejci, "MDI measurement in the IPTV", 2008, pp. 49–52, IEEE.

[79] R. Danchick and GE Newnam, "Reformulating Reid's MHT method with generalised murty k-best ranked linear assignment algorithm", in *IEEE Proceedings on Radar, Sonar and Navigation*. 2006, vol. 153, pp. 13–22, IEE.